TCG Infrastructure Working Group Reference Architecture for Interoperability (Part I)

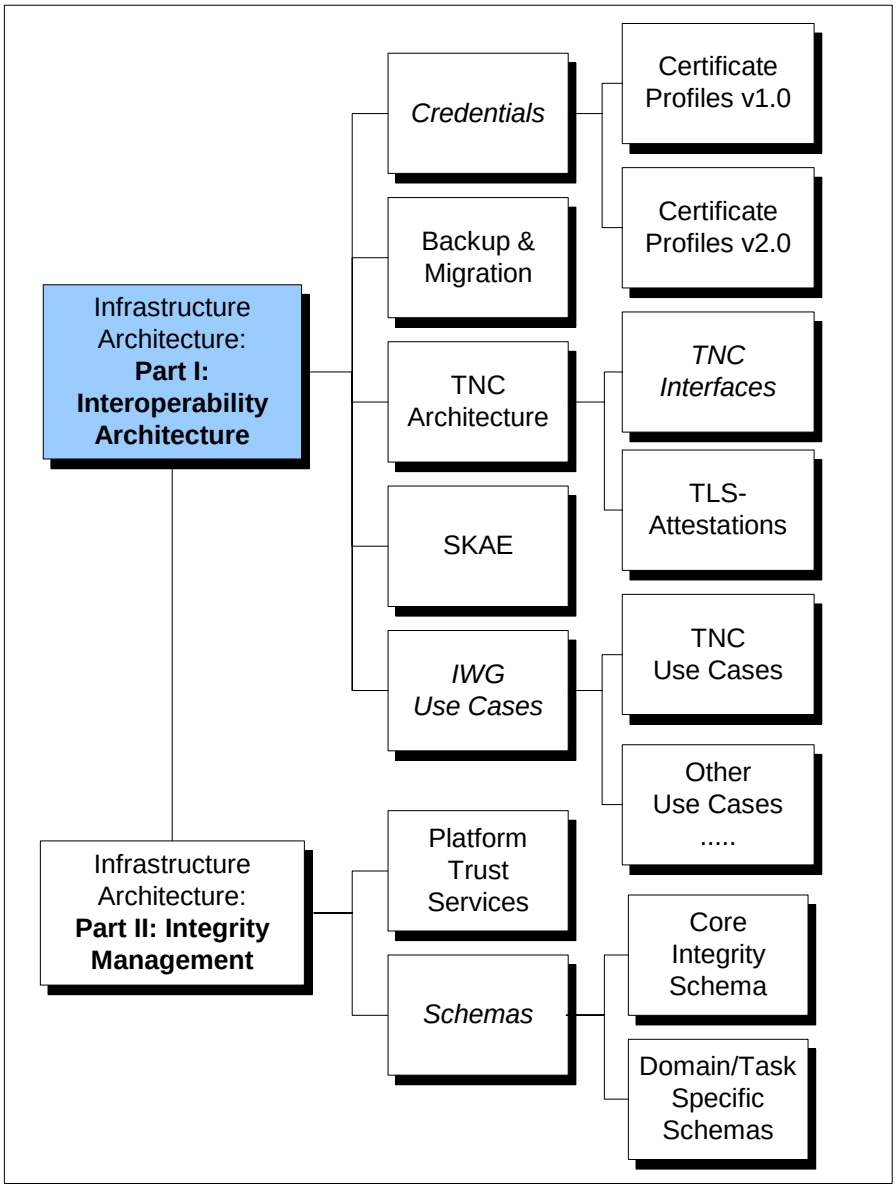
Specification Version 1.0
Revision 1
16 June 2005
Published

Contacts: techquestions@trustedcomputinggroup.org

TCG PUBLISHED

TCG

IWG Document Roadmap



Revision History

	Initial outline by Ned Smith. Document started by Thomas Hardjono.	3/15/2004
Rev 1.0_r0	Version 1.0_r0 submitted for 60-day internal TCG review.	11/24/2004
Rev 1.0_r1	Multiple corrections from various reviewers for Final version.	
	TCG Board of Directors approval for Version 1.0_r1 publication.	6/16/2005

Acknowledgement

The TCG wishes to thank all those who contributed to this specification. This document builds on numerous work done in the various working groups in the TCG.

Geoffrey	Strongin	AMD
Julian	Hammersley	AMD
Randy	Mummert	Atmel
Colin	Walter	Comodo
Mark	Redman	Freescale Semiconductor
Kazuaki	Nimura	Fujitsu Limited
Seiki	Shibata	Fujitsu Limited
Patrick	George	Gemplus
Boris	Balacheff	Hewlett-Packard
Graeme	Proudler	Hewlett-Packard
Matthias	Schunter	IBM
Diana	Arroyo	IBM
Lee	Terrell	IBM
Roger	Zimmermann	IBM
Andrew	Kegel	IBM
Markus	Gueller	Infineon
Johann	Schoetz	Infineon
Ned	Smith (IWG Co-Chair)	Intel Corporation
David	Grawrock	Intel Corporation
Monty	Wiseman	Intel Corporation
Ravi	Sahita	Intel Corporation
Stephen	Heil	Microsoft
Mark	Williams	Microsoft
Jennifer	Curtis	Microsoft
Hamid	Karimi	nCipher
Ari	Singer	NTRU Cryptosystems, Inc.
William	Whyte	NTRU Cryptosystems, Inc.
Andy	Cottrell	Phoenix
Mark	Schaeffer	Renesas Technology Corp.
Andrew	Nash	RSA Security, Inc.
Laszlo	Elteto	SafeNet, Inc.
Peter	Reed	SafeNet, Inc.
Michael	Willett	Seagate Technology
Robert	Thibadeau	Seagate Technology
Manuel	Offenberg	Seagate Technology
Brad	Andersen	SignaCert, Inc.
Nicholas	Szeto	Sony Corporation
Jeff	Nisewanger	Sun Microsystems, Inc.
Paul	Sangster	Sun Microsystems, Inc.
Thomas	Hardjono (Editor, IWG Co-Chair)	VeriSign, Inc.
Len	Veil	Wave Systems
Greg	Kazmierczak	Wave Systems
Lark	Allen	Wave Systems
Mihran	Dars	Wave Systems
Scott	Cochrane	Wave Systems

Table of Contents

1 Scope and Audience

The TCG Infrastructure Working Group (IWG) has defined a “reference” architecture aimed at existing and new infrastructure technologies having a goal of improving interoperability among systems containing TCG technology.

Architects, designers, developers and technologists who are interested in the development, deployment and interoperation of trusted systems may find this document helpful in providing both abstract and implementation specific insights for achieving interoperation between TCG-based systems.

2 Introduction

The purpose of this document is to provide a reference architecture for supporting environment around a trusted platform (containing TPM and TCG technologies), as defined by the TCG. The technology introduced by the TCG has garnered interest in the broader technology industry, as it has introduced fundamental concepts regarding hardware-based trust into mainstream computing. The relevance and importance of hardware-based trust is becoming increasingly apparent even to the average user, in the face of increasing threats to the users on the Internet. These threats range from viruses to identity theft, all of which have a direct impact on the user's daily life.

2.1 Inter-Platform and Intra-Platform Infrastructures

This document also represents the next-step forward for the TCG in defining how the TPM and its properties can be used to define and build a trusted platform used to interact with external entities. In reading this document, it is important for the reader to distinguish between the following general kinds of infrastructures, as the term "infrastructure" may have multiple meanings:

- Inter-Platform infrastructure: This term refers to the architecture and environment supporting the interaction between two (or more) independent platforms.
- Intra-Platform infrastructures: This term refers to the environment supporting the interaction between a TPM (within a platform) and other devices which may not be a TPM-based platform as defined by the TCG.

The current document refers to the first case, namely the Inter-Platform infrastructures. Although some aspects of the second case (intra-platform) have relationships with entities and functions within the IWG Reference Architecture, it is not currently the focus of the current document.

2.2 Layers of Abstraction

The notion of layers of abstraction is an important tool in providing an understanding of the role of hardware rooted trust in providing a basis for entities defined in the layers. The precise level of assurances obtained is very much dependent on the use-case deploying trusted platforms, and on which entities instances are defined over trusted platforms. The IWG is aware of the various possible layers of abstractions and entities that may "speak" (i.e. issue assertions) at the various levels in the layers. The layers are anchored on the trusted platform

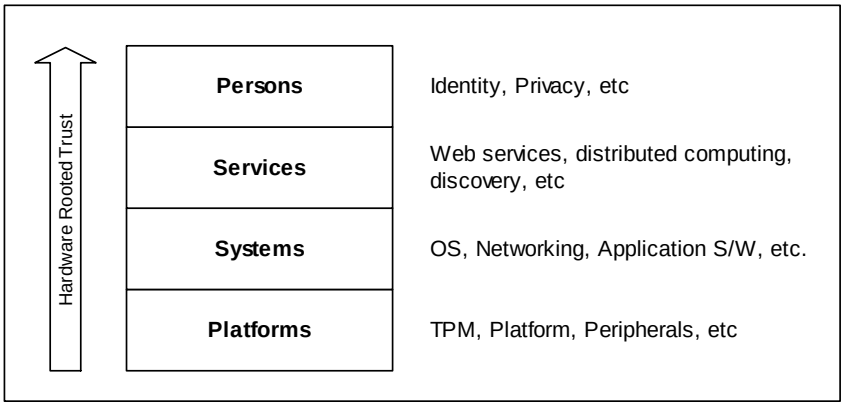


Figure 1: Layers of Abstraction

Figure 1 shows the general layers where assertion-generating entities may reside and where these entities may be viewed (at that level) as independent speaker of assertions. The identified levels provide a useful tool in categorizing functions, protocols and credentials involved in a given interaction between two (or more) TP-based entities.

Note that each the layers are build upon the layer beneath it, all based on the trusted platform. The layers identified are as follows:

- *Platforms layer*: The term "platform" refers to Trusted Platforms as defined by the TCG. Here, credentials are bound to the platform and all assertions made by the platform concerns the platform itself without regard or reference to high-layer functions and entities (e.g. OS, person).

  The platform layer is useful for scenarios or use-cases where a hardware-based platform is the entity being identified (e.g. in a two-party interaction) and where the assertions are made by the platform as an entity. For example, a TPM-based network device such as a hardware VPN gateway may be the entity to whom a VPN-client communicates and who issues assertions (about itself) to be consumed by the VPN-client. This network device can be seen primarily to be a platform-layer entity since it is a stand-alone device which lacks the richness of functions typically offered by PC operating systems.

- *Systems layer*: Here, the term "systems" refers to the group of software systems that may stand-up a single entity and generate assertions as a unique entity. These software range from the OS up to (one or more) applications software. Note that although the traditional operating system (OS) may be considered as part-and-parcel of a Trusted Platform, here it is included in the Systems layer to facilitate discussion with regards to hardware-rooted trust.

  The systems-layer is useful to express entities which contain a richer set of internal functions and which may offer a limited set of exported services/functions as part of a larger service, but whose service may be less meaningful or limited on its own. For example, a server sitting within a Grid-Computing Service (GC-Service) network may offer CPU processing power as a basic unit of service. A consumer of this service needs to obtain platform-level and system-level authentication before sending processing tasks to that entity. This server entity offers a very basic service and maybe limited to certain grid-computing related tasks. Its service is really a basic building block for a larger GC-Service, which is made-up of multiple entities of the same type and level. However, although it is only a component of a larger GC-Service, its security and TP integrity is crucial to the larger service as a whole. As a system (in the current IWG architecture context), it can make assertions regarding the trusted platform upon which is built and regarding the limited system functions it may offer.

  Note that a range of service types based on a given application can be grouped under the category of the system layer (from the perspective of a TP).

- *Services layer*: The term "services" here focuses primarily on those services consumable by external entities (inter-platform). Thus, although the term "services" may be used in different contexts (e.g. OS function), it is used here for inter-platform interactions where one TP-based entity is offering services to another. The term "services" is used to denote or highlight the fact that multiple systems (each based on a trusted platform) may make-up a given service as a whole.

  The web-services scenario provides a useful example to illustrate this layer. Consider an airline reservations service, which is made-up of multiple servers instances, each performing a subset of services (towards completing an airline reservation) and each of which is built upon a trusted platform. The end-user (consumer) may be concerned only that he or she is securely communicating with a legal entity offering a service, authentically identified through some service-level credential (e.g. corporate certificate) and that the front-end web-server is deploying a trusted platform. However, if the airline reservations company is in fact outsourcing portions of the service to other companies

(e.g. credit-card processing, hotel bookings, etc.), then it in-turn may insist that these other entities offer services running also on trusted platforms.

- *Persons layer*: This layer represents the human person an entity involved in one or more interactions with entities defined in the IWG architecture. The human person may be represented by a credential that is rooted in the hardware or platform credential (used by the human user). For example, the human credential could be signed using a (migratable) key which chained to an Identity Key (as defined by the TCG)..

Note that the above layers are intended to be a tool in identifying entities which interact at peer layer. Thus, it is constructive to view web-services (built on a trusted platform) to be communicating with each other at the services layer, while entities providing Identity Management Services may in fact communicate at the Persons layer.

2.3 Roadmap of IWG Documents

Aside from the current Framework Architecture for Interoperability document, there are a number of documents that are developed within the context of infrastructure. These are shown in Figure 2 in *italics*, with further explanations in Section 7.

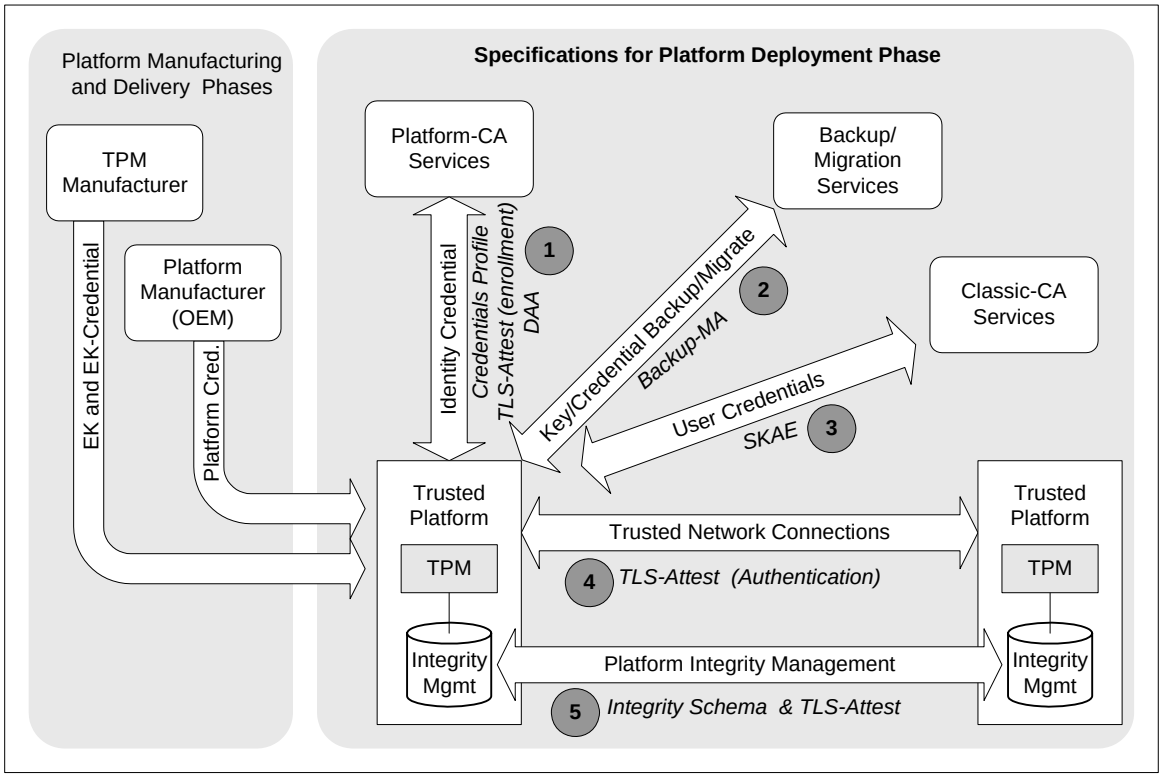


Figure 2: Specifications Roadmap Diagram

The current specifications being developed currently are briefly described as follows:

- *IWG Use-Cases*: The Use-Case document collects the set of important Use Cases that drive the definition of architectures and functions of an infrastructure supporting Trusted Computing. The document can be found in [1].

- *Credentials Profile*: The aim of Credentials Profiles specification is to collect, in one document, definitions for three of the credential types identified in the v1.1b TCG Main specification, namely, the TPM Endorsement (EK) Credential, the Identity (AIK) Credential, and the Platform Endorsement (Platform) Credential. The specifications can be found in [2]
- *TLS Attestations*: The TLS Attestations specification defines extensions to the TLS (SSL) protocol to convey integrity-related information from (to) a Trusted Platform. See document [6] for further details.
- *Integrity Management*: The purpose of the specification is to define platform integrity information, consisting of integrity assertions and integrity values, as required for a Trusted Platform. See document [4].
- *Backup/Migration*: The Backup/Migration document specifies the methods and protocol to perform backup of relevant cryptographic keys and data within a Trusted Platform, and the function of moving (migrating) keys from an “old” platform to a “new” platform. See [3].
- *Direct Anonymous Attestations (DAA)*: The DAA protocol allows the creation of an Identity Credential without a Platform CA, thereby allowing the Identity Credential to possess some anonymity properties. The DAA protocols is are briefly summarized in documents [8], [10], [11] and [22].
- *Subject Key Attestation Evidence (SKAE)*: The SKAE document specifies the use of evidence regarding a Trusted Platform in the event of enrolling for a user certificate from a Classic (Traditional) CA. See for [5] further details.

3 The Trusted Platform Lifecycle

Trusted Platforms (TP) provide a number of attractive security-features and capabilities. The notion of trust built upwards from a hardware-based root of trust provides levels of authentication of both platforms and users that previously did not exist. The complex nature of trust in the digital world necessarily demands infrastructures that support the provisioning, deployment and retirement of TPs, as platforms themselves have now become entities that are distinct from human users from a trust perspective.

3.1 TP Lifecycle

In the current section we introduce the notion of "infrastructure" as the set of entities, functions and roles that are needed to support the use of Trusted Platforms throughout their lifecycle. For simplicity of understanding, these supporting entities and functions are categorized into three (3) broad infrastructures, as shown in Figure 2. Note, however, these three infrastructures share many common aspects and may be implemented by the same entities. Also, platform recycling may involve a return (repeat) to earlier phases in the lifecycle (e.g. old TP with new EK-key, etc).

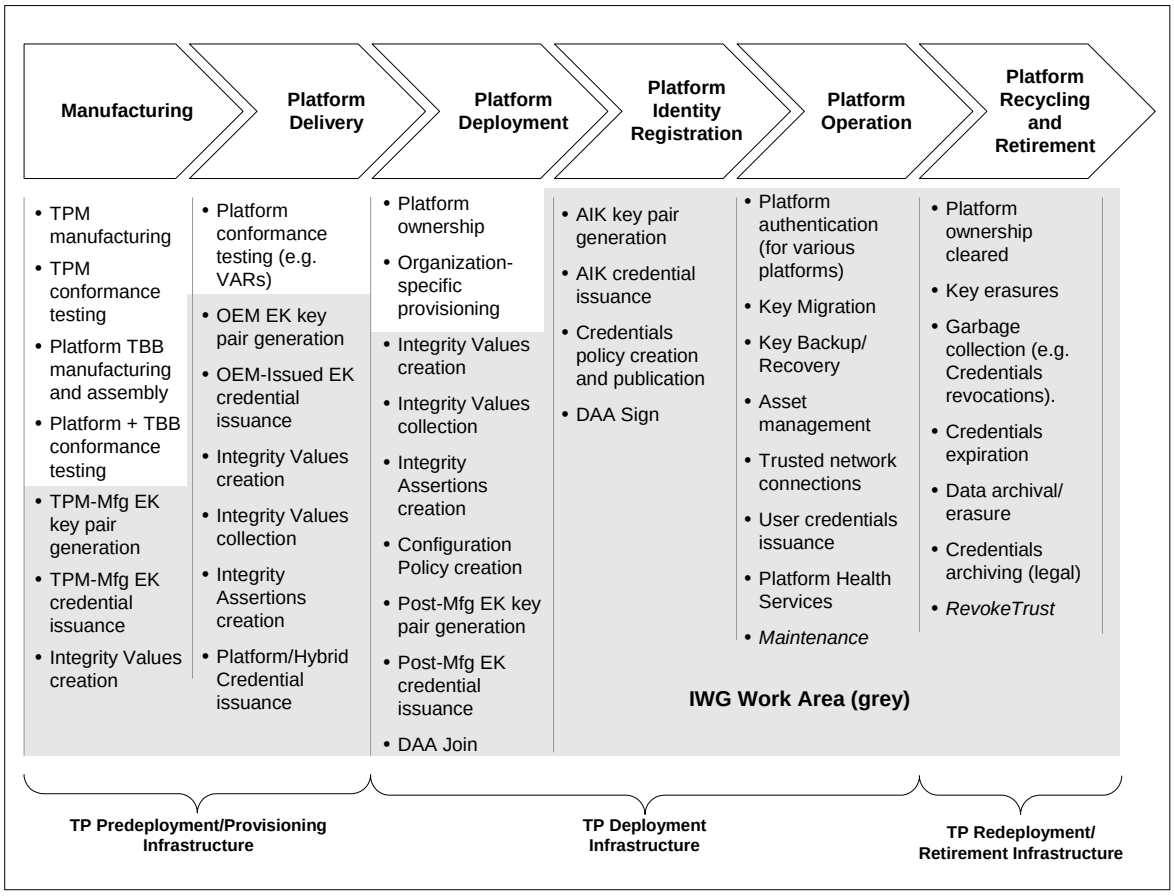


Figure 3: The Trusted Platforms Lifecycle

3.2 Three (3) Categories of Infrastructures in the TP Lifecycle

The three broad categories of infrastructures supporting the Trusted Platforms Lifecycle are intended as a tool to classify aspects and features of the TP lifecycle. As such, they should be seen as a grouping of functions and entities that are involved in a particular phase of a TP's life.

These three infrastructures are as follows:

- *TP Predeployment Infrastructure*: These are the set of entities and functions that support the preparation of TPs before they are deployed. Some examples include entities that provide functions supporting the creation of EK-credentials (early and late), Validation-credentials and other (pre-AIK) credentials, and those performing conformance-related functions. Specific entities involved in the TP pre-deployment infrastructure are TPM manufacturers, the motherboard suppliers that connect the TPM, TBB, and physical presence signal to the platform motherboard, system builders (OEMs, ODMs, and white box makers), as well as compliance testing laboratories hired by these other entities.
- *TP Deployment Infrastructure*: These are the set of entities and functions that support the actual use of Trusted Platforms outside the manufacturing control boundary. Typically, the Ownership of the platform has been established, and one or more Users are using the platform. Example of entities in this infrastructure category include AIK-credential issuing authorities (e.g. Platform-CAs), Authentication Servers supporting platform authentication, Policy Servers supporting TP-aware IT strategies, and others such as Value Added Retailers.
- *TP Retirement/Redeployment Infrastructure*: These are the set of entities and functions that support the retirement (de-provisioning) of existing TPs in the case of old systems and the re-deployment of existing TPs with a new (fresh) set of credentials, possibly with new Ownership.

It is important to note that the infrastructures categorization does not imply that the entities and functions in each infrastructure require distinct implementations or embodiments. Thus, it is possible that in reality a single entity supports multiple functions across two or all three categories of infrastructures.

In general, the boundary between the Pre-deployment Infrastructure and Deployment Infrastructure is crossed when an entity takes possession of the physical TP and performs the take-Ownership operation on the TP. The boundary between the Deployment Infrastructure and the Retirement/Redeployment Infrastructure consists of the combination of (new) Ownership and key/credential erasure.

3.3 Lifecycle Phases

Following the phases illustrated in Figure 3, each of the phases are described below from the perspective of infrastructure functions, entities and services involved in a given phase. Note that some functions and roles may be valid across phases, and in some cases may even be repeatable in two or more phases (adjacent or non-adjacent phases).

3.3.1 Manufacturing

The Manufacturing phase covers the manufacturing and assembly processes involved in the creation of a trusted platform as understood by the TCG. This includes TPM hardware manufacturing, Trusted Building Block (TBB) components and motherboard manufacturing, and the process bringing together all the hardware components defining a trusted platform. In

addition, this phase covers the various conformance testing that has to be performed on the TPM, TBBs and TP as a whole.

From the infrastructure perspective a number of important functions occur at this phase:

- *Integrity Values Creation*: Information regarding a given TPM, TBB components, Firmware, Software and Trusted Platform configuration must be collected and made accessible for input into the next phase. This information must be generated in this phase by the manufacturers of each component, with the aim of being consumable by Conformance Laboratory who verify the correctness of the implementation of a given TPM, TBB, Platform or any of its hardware and software components.
- *TPM-Manufacturer EK key pair generation (Early/Normative)*: A TPM manufacturer is expected to make the EK key pair physically present inside a TPM during this phase. Typically, the TPM manufacturer generates the EK key pair and inserts it into the TPM Platform Credential issuance prior to delivering the platform to its owner. There are two approaches to key generation and insertion; 1) generate keys off-chip and insert as part of TPM construction, 2) generate keys on-chip. The first approach is presumed to be performed by a TPM manufacturer, while the second can be performed by anyone having physical access to the TPM/platform during the manufacturing process. In order to distinguish the normative EK key pair generation in this phase from that in other later phases, here it is also referred to as *early* EK key pair generation. Early EK generation is the normative behavior.
- *TPM-Manufacturer EK-credential issuance (Early/Normative)*: A TPM manufacturer is expected to issue an EK-Credential during this phase for TPM devices it manufactures. This is the normative behavior. Note that EK-Credential issuance must come after EK key pair generation but not necessarily immediately following. In order to distinguish the normative EK-Credential issuance in this phase from that in other later phases, here it is also referred to as *early* EK-Credential issuance.
- *TPM-Manufacturer DAA-Credential issuance*: A TPM manufacturer could issue a DAA-Credential by executing the DAA-Join Protocol. A TSS would be temporarily required, which output (DAA-Credential) must be able to be imported into the owners TSS later.

3.3.2 Platform Delivery

In this phase, a platform is in the process of being delivered to an Owner, though not yet in the physical possession of the intended Owner. This phase is closely tied to the previous phase, as much of the integrity-related information produced by manufacturers in the previous phase must be collected, evaluated and represented as integrity assertions.

It is important to note that Platform conformance testing and evaluation is shown to occur in this phase, the intention being to denote that fact that a number of Integrity Assertions are indeed created as a result of Platform conformance testing and evaluation. Note that similar to other new technologies, the evaluation of a Platform can occur after product shipment due to the fact that an evaluation process may take many months and possibly years. Conformance testing may occur as part of product release but would likely not be fully applied until after the evaluation. Integrity values creation is thus largely the responsibility of manufacturers and VACR (value added content providers).

Additionally, it is worth noting that although the preceding Manufacturing Phase includes both *TPM Conformance Testing* and *Platform and TBB Conformance Testing*, the Integrity Values creation and Integrity Assertions creation corresponding to those actions can occur also in the current phase (and the next). This is because the credentials issued during the first three phases may contribute to the body of integrity values and integrity assertions.

From the infrastructure perspective a number of important functions occur at this phase:

- *Integrity Values Creation*: During the Platform Delivery phase, the platform manufacturer (i.e. OEM) must generate integrity values pertaining to the platform, the TBB components, Firmware and Software that make-up the platform. In addition, a Trusted Platform configuration may also be defined by the OEM prior to shipment to the customer. Thus, within this phase the set of integrity values increases from the previous phase and will be input to the Integrity Values collection process.
- *Integrity Values collection*: The integrity values generated by the manufacturer of the Platform may be collected at this stage. Depending on the exact platform manufacturing process, different collection mechanism (e.g. file, website, CD) may be employed by manufacturers and Conformance Evaluation entities for each of the components of a platform.

  Note that not all integrity values may be of interest to a given Conformance Evaluation entity. Thus, for example, a lab evaluating a TPM chip may only be interested in integrity values pertaining to a given TPM from a given TPM-vendor, employing the collection mechanism agreed upon with that manufacturer. Similarly, a Platform Conformance evaluation lab may require the OEM to supply it with all the integrity values it needs to evaluate the platform, leaving the OEM to collect the component integrity values from the various sources.
- *Integrity Assertions creation*: Conformance Evaluation entities publish their positive findings regarding the evaluation of a given TPM, TBB component and Platform in the form of Integrity Assertions in a manner that preserves the fact of their publication (e.g. digitally signed). Manufacturers, OEMs, VARs, IT departments and independent labs reasonably may function as conformance evaluation entities. These assertions are intended to have clear semantics, and can be represented syntactically in a number of forms including X.509 certificates, XML certificates, XML files, text files, and other representations.
- *OEM EK key pair generation*: EK key pair generation (on the TPM) can occur in this phase, prior to the platform being taken over by its Owner (in the next phase). This is to denote the possibility that an EK key pair be generated by an entity that it is *neither* the TPM Manufacturer nor the Platform Owner (i.e. entity chosen by the either the Manufacturer or the Platform Owner). EK key pair generation during this phase is regarded as *early* EK generation (non-normative).
- *OEM EK-credential issuance*: EK-credentials can similarly be issued by an entity that is in an authoritative position to make assertions about the validity of an EK key pair. Such an entity need not be *either* the TPM Manufacturer or the Platform Owner. Ostensibly, EK-credential issuance may be done by an entity that did not generate the EK key pair, but not all issuers can speak with the same authority. EK credential issuance during this phase is regarded as *early* EK credential issuance (non-normative).
- *OEM DAA-Credential Issuance*: see above.

3.3.3 Platform Deployment

The start of the Platform Deployment phase is signified by the take-ownership of the platform by its Owner, with physical presence. This event is significant because for the first time since platform manufacturing the platform is outside the control of the various contributing manufacturers.

Platform Ownership signals the start of a number of functions that pertain to deployment of the platform The Owner of the platform should be physically present with the platform in order to “activate” it and issue “take ownership” commands. Taking platform ownership may be accompanied by setting of one or more passwords. The platform owner is typically the IT

administrator in the case of an Enterprise, while in the case of a consumer purchased platform it is the end user.

From the infrastructure perspective a number of important functions occur at this phase:

- *Integrity Assertions collection*: The Owner of the platform needs to collect the Integrity Assertions regarding components of the platform from the entities that produce and/or tested those components. The Integrity Assertions statements found in certificates or manifests assert that certain properties of a given component hold (See Reference [4] for exact semantics of the Integrity Assertions). For example, assertions can describe manufacturing processes followed, root of trust designation and semantics of EK creation.
- *Post-Manufacturing EK Key Pair generation (Late)*: When a EK key pair is generated and made physically present inside a TPM during this phase it is referred to as *late* generation. If Late EK key generation occurs in the Platform Deployment phase, it is the platform Owner (e.g. IT administrator) that performs key pair generation operations. Late EK generation can be problematic because it prevents EK credential issuance during manufacturing phases. Hence, platforms having late EK generated keys may have diminished value outside the owner controlled domain.
- *Post-Manufacturing EK Credentials Issuance (Late)*: When an EK Credential is issued after take-ownership of a platform is performed, then the process is referred to as *late* EK Credential issuance.

  When Late EK-Credential issuance occurs in the Platform Deployment case, it is the platform Owner that makes the decision as to who issues the credential. Credential issuance could be done by the Owner (e.g. IT administrator) or by some entity trusted by the Owner. However, the chosen issuer claims may not be authoritative for all entities that seek to establish trust in the platform.

  It is important to note that although there may be some logical continuity between late EK Key Pair generation and late EK Credential issuance, the two processes need not necessarily occur in the same phase in the TP Lifecycle. That is, it is permissible that EK Key Pair generation occur prior to the Platform Deployment phase and for the EK Credential Issuance to occur in the current Platform Deployment phase.
- *Platform Credentials Issuance*: The platform will contain one or more credentials which are information to the identification and trust properties of the platform. One of the important tasks at this phase is the issuance of the Platform Endorsement Credential, which attests to the uniqueness of the TPM instantiation on the platform. Additional credentials may attest to the binding of the TPM to the trusted platform containing trust properties and conformance to industry standard security specifications. It is here that the Integrity Values produced by the previous phase and the current phase becomes important to the evaluation of the platform.
- *Configuration Policy creation*: The Owner of the platform creates policies expressing the acceptable configurations of platforms in the domain of the Owner. There are a number of possible aims for this action, depending on the specifics of each domain configuration and the use case. For example, the policy may be reflected in Authentication Servers that verifies a client platform configuration as part of the authentication process for network connectivity requests. Alternatively, the configuration policy may be part of the IT asset management approach in which the network presence of each platform is detected and monitored. Other examples of the use of the policy can be found in the IWG Use-Cases document [1].
- *DAA-Join*: Since the amount of trust accorded to a Privacy-CA may be too much for certain areas of application, it is sometimes desirable to obtain an identity-credential without a dependency on a Privacy-CA to mask-out PII-related information. Thus, an alternative to obtaining an AIK-Credential from a Privacy-CA is to obtain a DAA-

Credential using the DAA-Join protocol from a so called DAA-Issuer. This step is distinct from the DAA-Sign step in which the platform proves possession of a DAA-Credential and at the same time can authenticate an AIK (see Platform Identity Registration Phase). Note that in practice the DAA related functions can only occur after platform ownership has been established.

3.3.4 Platform Identity Registration

The next phase in the TP Lifecycle is the establishment of the so called "identity credentials", which broadly speaking is the assignment of a certificate that "speaks" on behalf of the trusted platform. A TP would use an identity-certificate (i.e. AIK-Credential) to assert to the world that it is a Trusted Platform (TP) conforming to the definition of a TP as specified by the TCG. Note that an AIK-Credential does not carry TP-specific information that can unambiguously distinguish one platform from another. Thus, the AIK-credential asserts that a given platform is a TP, but does not permit multiple AIKs to be correlated.

- *AIK Key Pair Generation*: The generation of the AIK key pair occurs in this phase. The Owner of the TP can generate the key pair.
- *AIK-Credential Issuance*: In the TP Lifecycle the entity that issues an AIK-Credential is referred to as the *Privacy-CA,* as a form of a Platform-CA. The Privacy-CA is trusted to correctly evaluate Integrity Assertions and the Owner-specific policies as input into the process of issuing AIK-Credentials. It is also trusted to keep the link between the EK and AIK private.

  The Privacy-CA in practice can be a local AIK-Credential issuer (e.g. Enterprise IT Administrator) or it can be a public certificate authority in the sense of a Classic CA. In either case, the requirement holds true that no platform-identifying information should be carried inside the AIK-Credential.
- *DAA-Sign*: The DAA-Sign function occurs in the current phase and can be seen as a continuation of the DAA-Join step in the previous step. Note that both DAA-Join and DAA-Sign can in fact occur in the current phase if the circumstances demand. In the DAA-Sign the platform interacts with the Verifier in order to convince the Verifier that the platform is genuine, as previously established (by the DAA-Issuer through DAA-Join) in the previous phase. In other words, it proves possession of a DAA-Credential and at the same time can authenticate an AIK.

3.3.5 Platform Operation

Once a platform has been configured by its Owner, and the platform's user has one ore more identity credentials then it is ready for deployment in various use case scenarios. One important fundamental operational support that needs to be provided to a platform is Backup/Migration of keys. This function is truly an infrastructure function (like obtaining identity credentials), as it is crucial to the sustainable usage of the platform.

- *Migration and Backup/Restore*: Since cryptographic keys play an important role in the proper functioning of a TPM and a Trusted Platform, its availability is crucial to the day-to-day use of a platform. As such, back-up of these cryptographic keys and certificates are important in the face of possible hardware and other system failures. Due to the sensitivity of the keys and certificates, a secure backup procedure must be employed to protect against theft and loss of those keys. In addition, the migration of the keys and certificates (and user data) from an old platform to a new platform is necessary to ensure that the end-user can continue to gain access to data and applications in the new

platform. The IWG Backup/Migration document [3] provides the TCG specifications for Backup and Migration.

- *Platform authentication*: One of the primary uses of integrity information regarding a platform is for the authentication of one platform by another. Platform authentication can occur at various levels of the service abstraction, but always involves the reporting of the integrity status of a platform (the Requestor) to another (the Verifier). This reporting must be done in a secure manner, as part of strong mutual-authentication protocol. See Reference [6] for further information.
- *Trusted Network Connection*: A particular instance of platform authentication is that occurring at the Network layer. Here the intent to use platform integrity information to perform "device" authentication within the context of the 802.1X Authentication Framework, driven by policies governing which platform integrity information is exchanged, device-level access policies and user access policies. The integrity information also includes information regarding security-specific applications (e.g. virus versions, patch versions, etc) and is consumed by various network-level services (e.g. VPN-gateway, Firewalls, etc). Document {TNC-SPC} provides the TCG specifications for Trusted Network Connections.
- *User Credential Issuance*: User certificates have become a day-to-day feature of communications in the Internet, including secure messaging (e.g. email, S/MIME), web-transactions and VPN access. Typically, user-certificates are obtained from a Classic Certificate Authority (Classic-CA), either in a public or private/closed capacity. When a user enrolls for a user-certificate to a Classic-CA, the integrity information regarding the user's platform can provide a higher level of assurance to the CA regarding the origins of the certificate-request. In addition, the CA can encrypt the newly-issued user-certificate (and possibly the key-pair) to the user's platform as a target (i.e. decipherable only on the same platform). Document {SKAE} provides the TCG specifications for trusted user-certificate enrollment based on trusted platforms.
- *Asset Management*: Another instance of the use of platform integrity information is for IT management to perform asset tracking and management for all platforms under his/her administrative jurisdiction. Here the context is similar to Trusted Network Connect in the sense that the IT Administrator could only allow onto the network machines that have successfully completed platform authentication. In this alone, the IT Administrator can glean information about the presence of platforms on his/her network. However, a strong case of asset tracking can be established by each platform reporting their current configuration, including all the software installed on the platform, anti-virus signature files, patch versions, hardware driver versions, and so on. And thus, platform integrity information provides a possible wealth of information to the IT Administrator (who is the Owner of all the Enterprise's platforms) for the purpose of hardware/software asset management.

- *Platform Health Services*: Related to Asset Management is the Platform Health Services, which is a set of services that can evaluate the status of the integrity of a platform against a set of policies regarding that platform. The health of a platform should cover the operational-relevant aspects of the platform (e.g. Backup is due, AIK-credential still valid), as well as aspects that are use case specific (e.g. Software license expired, peripheral hardware has new driver).

3.3.6 Platform Recycling and Retirement

Similar to the current PC lifecycle, it is expected Trusted Platforms will make their way into secondary (recycled) markets. Unlike ordinary computing platforms today which can retired by simply removing their hard-disks or erasing them, retiring a Trusted Platform containing a TPM

requires care in ensuring that important keys and certificates are removed from the platform before it is retired:

- *Platform ownership clear*: Since the Owner of a platform has control over aspects of the trusted platform, prior to retirement or recycling of a platform its current Owner must ensure that he/she clears the TPM of the current keys, certificates and other parameters. This can be done using the *TPM_ClearOwner* function in the TSS.
- *Key erasures*: Besides clearing the TPM of Owner-specific keys and certificates, the Owner must also erase keys and certificates which belong to or were created by users of the platform. This is to ensure that those keys and certificates are not accessible to other persons who are unauthorized to use or access them (e.g. new Owners of the platform). If keys and certificates belonging to users are still relevant (e.g. used to seal user data), then they must be backed-up or migrated to the relevant new platform.
- *Garbage Collection*: Here, garbage collection generally refers to the proper management of sensitive parameters and information that may reside on the TPM, or is in someway tied to the TPM (e.g. data encrypted using keys stored in NV-Storage). One important example is that of revoking certificates that may be unusable after the platform ownership is cleared. Although the private-keys corresponding to certificates may be erased automatically through the ClearOwner function, it is good practice to inform the Issuer of certificates that a certificate is no longer in-use. This allows the Issuer to remove the certificate from its active-certificates list and publish the revoked certificate serial number in either a CRL or through an OCSP server.
- *Credentials Expiration*: The issue of credential expiration is important in the context of platform retirement and/re recycling. The problem is particularly relevant when a given credential for a use-case (e.g. self-signed user certificate) is *chained* to a platform-related credential (e.g. AIK-credential). Thus, although the platform-related credential may be erased (or requested to be revoked) by the platform Owner, the credentials chained to these (revoked) platform credentials may have a longer expiration-time. This implies that the Owner or user may also need to revoke the chained credentials. This is the classic problem of certificate path validation.
- *Data Archival/Erasure*: Although data archival or backup is an obvious task to perform prior to retiring or recycling a platform, in the case of a Trusted Platform data may be sealed in a number of ways, with the corresponding keys either sealed or stored inside the TPM (e.g. NV storage). Thus, in archiving data it is paramount that the keys which encrypt the data are appropriately extracted and/ore migrated with the data to the backup platform. Here, the TCG Backup/Migration Protocol (see below) can play a role.
- *Credential Archiving*: Some legal requirements dictate that corporations must archive outgoing (and incoming) emails and other digitally-signed documents. Thus, although some credentials may not be in-use any longer, they need to be archived for some possible future need (e.g. to re-verify signed email and documents).

4 TP Deployment Infrastructure

In the current section we provide some fundamental concepts pertaining to the notion of authentication in the context of Trusted Platforms (TP). First we explain the basic three-entity authentication model for the establishment of trust among the entities. Although the context of discussion is trusted platforms, the intent of the model is to be applicable to other contexts and use cases that are built on the notion of trusted platforms. To that extent the current Section discusses the relationship between the basic model with the layers of abstraction (introduced in Section 2.2) and the relationship of trusted platform with user authentication. This Section also provides some background regarding the classic Four-Corners model which can be seen a superset of the basic model followed in this Architecture.

4.1 Basic Model for Platform Authentication

The starting point for the authentication model underlying the TCG Infrastructure architecture is the three (3) basic entities shown in Figure 4. These are the *Requestor*, *Verifier* and the *Relying Party*. These three entities capture the basic concepts in the TCG (such as attestation by the platform) and at the same time they also reflect the traditional model for entity authentication. Furthermore, they map readily into many Use Cases which have been captured by the IWG.

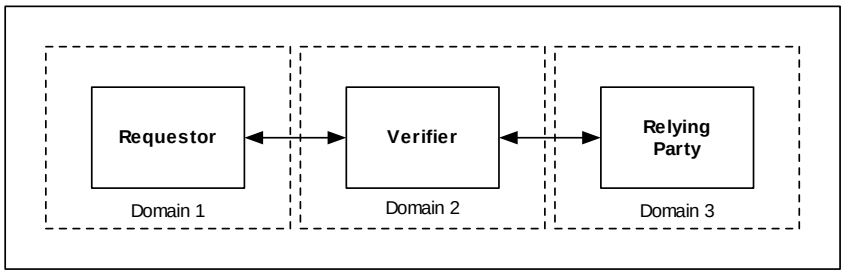


Figure 4: Basic Model for Platform Authentication

In this basic model the Requestor is performing a transaction with the Relying Party through the mediation (direct or indirect) of the Verifier. The Relying Party relies on the Verifier to evaluate the *assertions* or claims presented by the Requestor. The Verifier performs the evaluation of the Requestor's assertions based on some set of criteria or rules, which and understood by all three parties and have been established through some out-of-band method. It is important that all three parties understand the same criteria (both syntax and semantics) in order for all three to communicate meaningful assertions.

The outcome of the Verifier's evaluation of the Requestor's assertions can be binary (True or False), or it can be a score (within a range of values) based the agreed criteria for evaluation. The notion of a "score" is intended to reflect the fact that many transactions in the real world have results that cannot map easily (or even logically) into a binary value. Often, a Verifier can only afford to offer a score value to a Relying Party, where the final decision resides with the Relying Party.

Note that the model above accommodates an interpretation in which the Requester is a human user, and where the Verifier performs an authentication of the user based on some user credential. Thus, trust that is rooted in hardware could be extended through transitive trust relationships, through the human Owner of the platform, ending in the end-user that is trusted by the Owner to use the platform.

Figure 4 also introduces the notion of *domains*, which captures the basic understanding that realizations of these entities may reside under differing jurisdictions of control. Examples of such jurisdictions include administrative domains, security domains, legal domains, networks,

geographic locations and others. The use of domains also indicates the need of the three entities to use the same (or compatible) semantics (and preferably the same syntax) to express assertions and evaluation results, as well as policies and meta-policy information.

One of the primary aims of the model is to be simple and flexible in order to traverse all the layers of abstractions (Section 2.2), be applicable within each layer and allow cross-layer trust relationships to be established. Within each layer of abstraction some examples of the use of the model are as follows (Figure 5):

- At the Platforms layer the basic model of the architecture is applicable to cases involving TP-to-TP mutual authentication using the attestation-by-the-platform approach, where the Requestor and the Relying Party are Trusted Platforms and where the Verifying could be the Platform CA.
- At the Systems layer, an example of the model's applicability is the network end-point integrity where the Requestor is an 802.1X Supplicant, the Verifier is the Radius Authentication Server, and where the Relying Party is the 802.1X Authenticator entity (e.g. switch).
- At the Services layer, an example of the model's applicability is EDI in the web-services context, where the Requestor and Relying Party are web-service providers (describing their services using WSDL), and where the Verifier could be a UDDI provider.
- At the Persons layer, the architecture's basic model maps quite readily into the classical four-corners financial transactions use case where the Requestor is a human person seeking to use his or her credit card for a transaction, the Relying Party is the merchant and the Verifier is the Bank working on behalf of the merchant (and is possibly the issuer of the person's credit card). In addition, cross-layer trust can be established between a Person and the trusted platform through appropriate user authentication methods.

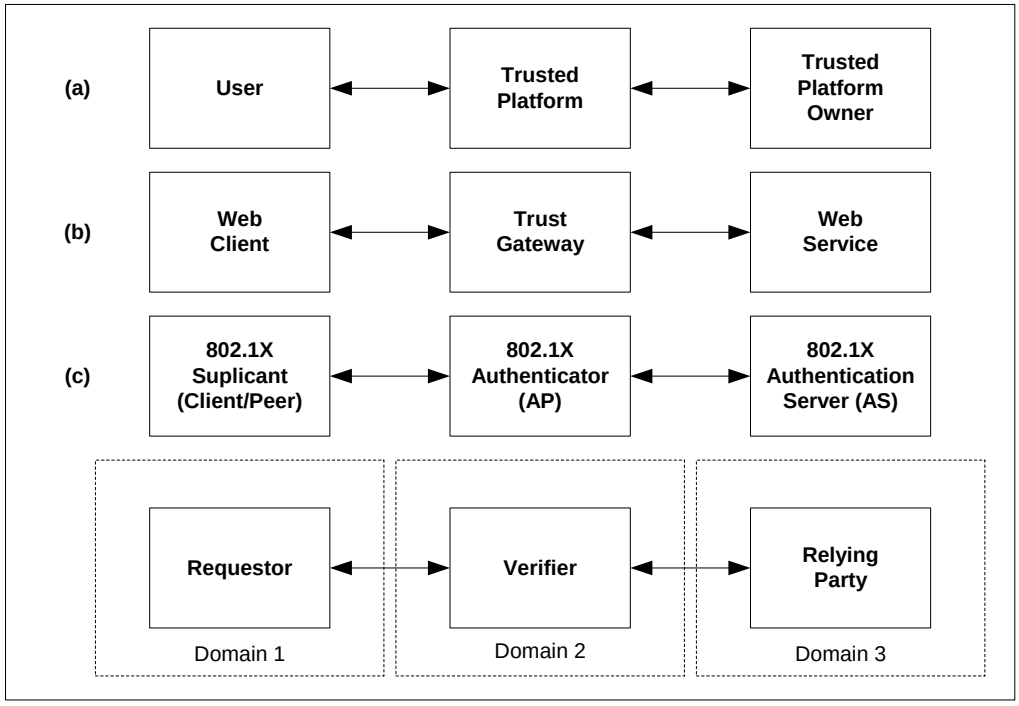


Figure 5: Example of mapping of the Basic Model to Layers of Abstraction

4.2 User Authentication and Trusted Platforms

Another dimension of the Basic Model for platform authentication is the use of trusted platform for user authentication in the context of that user seeking certain services or access to resource. The three-entity model introduced above – namely with the Requestor, Verifier and Relying Party – maps readily into the case of user authentication. Figure 6 attempts to show this relationship.

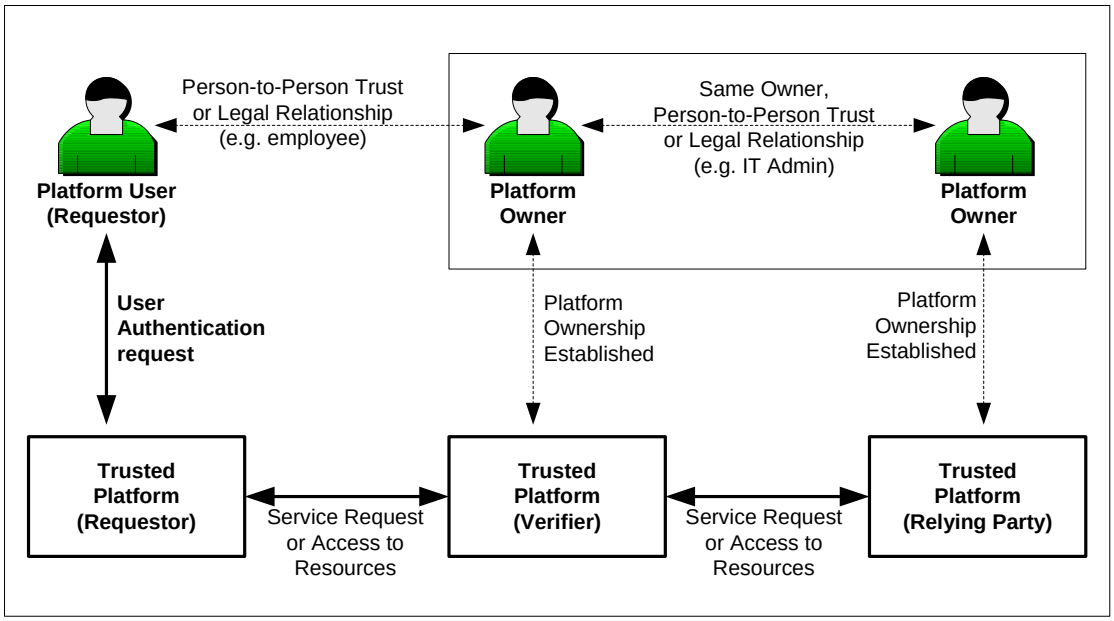


Figure 6: User Authentication using Trusted Platforms

As explained in the TP Lifecycle discussion (Section 3), prior to TP being deployed an Owner of the platform must take ownership of the platform through methods defined in other TCG specifications (see References [10] and [11]). Part of defining Ownership of the platform is the definition of the *User(s)* of the platform. Note that for some use case, the Owner maybe the sole user of the platform.

Consequent to the definition of Users of a given platform is the need (during platform operation) of the User to authenticate itself to the trusted platform via some user-credentials recognized by the platform.

In Figure 6 a User seeks to access resources or obtain services at a local platform or a remote platform. Trust relationships have been established prior to the platforms being deployed, notably between the platform Owner and the User, and possibly between the Owners of the local and remote platforms. The key feature of this diagram is the fact that the User must authenticate herself or himself to a Trusted Platform, with the platform being the Verifier and the resource or service entity (possibly also a Trusted Platform) being the Relying Party.

The figure illustrates the flexibility of the Basic Model for authentication shown previously in Figure 4 and the fact that the model maps readily into other entity configurations, in this case for User authentication.

4.3 Overview of model components

An important concept that distinguishes the above basic model from one that is relevant to trusted computing is the notion of a trusted platform containing a TPM that features *protected*

capabilities, integrity measurement and storage, and *integrity reporting*. All three properties or functions are core to trusted computing. These properties of trusted computing are reflected in Figure 7, which captures a more detailed view of a TC-centric architecture and its components.

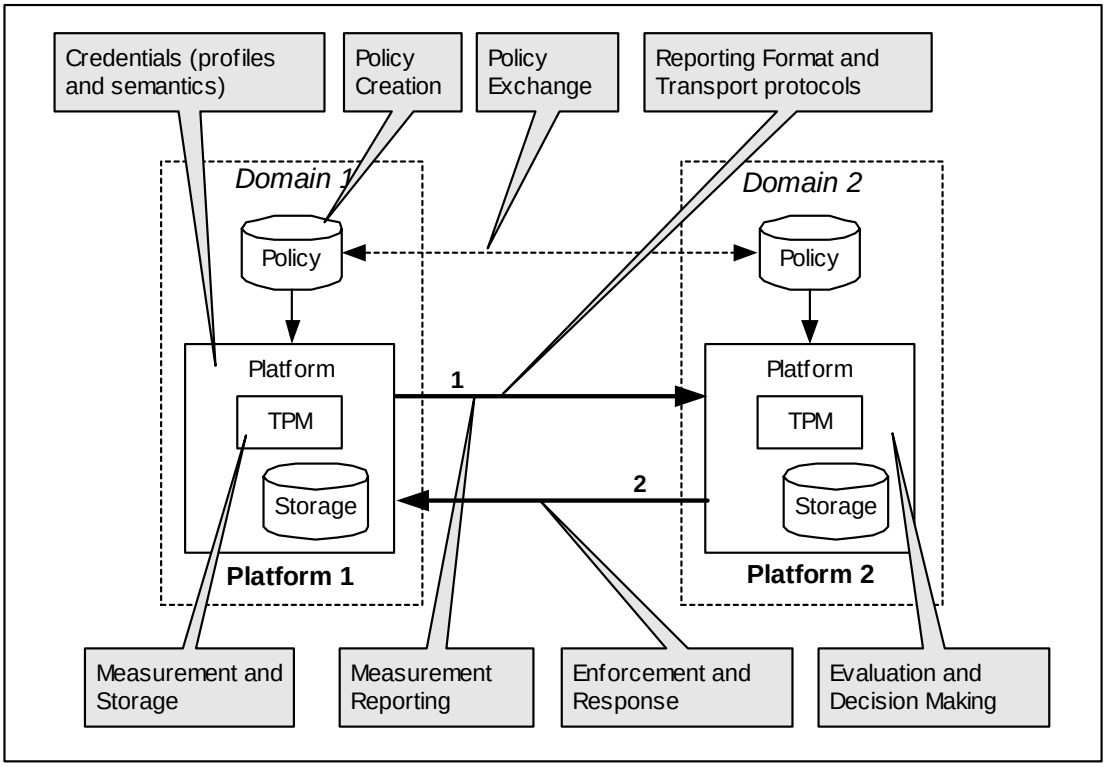


Figure 7: Overview Components

These components are as follows:

- *Credentials and Profiles*: a Trusted Platform has a number of credentials, including EK-certificate, TPM Conformance-certificate, Platform-conformance-certificate, one or more AIK-certificates, and one or more Validation credentials. The profiles for these certificates will be defined, together with the precise meanings/semantics and purpose of each field, in order to provide maximum interoperability across platforms, layers and service providers. The IWG credentials document [2] addresses these topics.
- *Measurement and Storage*: Integrity measurement is the process of obtaining metrics of platform characteristics that affect the integrity (trustworthiness) of a platform; storing those metrics; and putting digests of those metrics in PCRs. An intermediate step between integrity measurement and integrity reporting is *integrity storage*. Integrity storage stores integrity metrics in a log and stores a digest of those metrics in PCRs.
- *Measurement Reporting*: Integrity measurement reporting is the process of attesting to the contents of integrity storage. In the current context, reporting is relevant for two end-points that wish to asses their respective platform trustworthiness.
- *Reporting Formats and Transport Protocols*: When integrity measurements need to be communicated between two end-points based on trusted platforms, a suitable format for the measurement values needs to be standardized for interoperability. In addition, transport mechanisms needs to be identified or defined, for each area of application (e.g. web-services, network end-point integrity).

- *Evaluation and Decision Making*: When a platform seeks to asses the integrity trustworthiness of a second platform with whom it is communicating, it needs to evaluate that second platform based on some policies which are meaningful and actionable by both platforms. The outcome of platform evaluation is not limited to binary results (such as success/fail), but may include ranges of values (e.g. 1 to 100) indicating the level confidence the evaluating platform has with regards to its assessment. Note that the outcome of an evaluation process by an evaluating platform may be consumable by a third party who must understand the semantics of the evaluation result coming from the evaluating platform.
- *Enforcement and Response*: Depending on the exact configuration of an evaluating platform, the platform may in fact be a *policy enforcement point* (PEP) for a given set of environmental-specific policies. In addition, the platform may return *responses* to another platform, of whom it evaluated.
- *Policy Creation and Management*: In order for two (or more) platforms to interact with assurance of each other platform type and configuration, policy at each end-point must be created and managed throughout the life cycle of the platforms.
- *Policy Exchange*: Interactions between trusted platforms must be governed by policies of the respective domains within which the platforms reside. The need for interoperability of platforms across domain boundaries implies that policies written for both attestation-of-the-platform and attestation-by-the-platform need to be aware of the capabilities of respective platforms, and that such policies need to be communicated or exchanged across domain boundaries.

4.4 The Four Corners Model: Historical Perspective

The model underlying the current IWG Reference Architecture in this document has some historical precedent in the form of the Four Corners Model of interaction between a Requestor and Relying Party (Figure 8).

In this Figure the labels inside the boxes represent conceptual functions relating to trust establishment. The TCG-labels outside the boxes represent the TCG entities in the current document, while the labels in brackets in the inner part of the diagram denote real-world entities from an example in the financial world.

Conceptually, the model attempts to capture interaction between two entities (Clients), each of which have a trust relationship with a *Trust Service* provider. The Trust Service provider issues security assertions and evaluates security assertions regarding the Clients. Thus, in Figure 8 Trust Service A stands behind assertions it issues regarding Client#1, whereas Trust Service B performs assertion evaluations on behalf of Client#2. One key assumption here is that there must be some existing trust relationship between the two Trust Service providers before any transaction can occur between Client#1 and Client#2. This model is useful because it captures some fundamental behaviors of the Clients (e.g. people, institutions) in the real world, and the model can be mapped to a simpler 3-corner case or expanded to address additional entities (each of which correlating to one or more of the four corners).

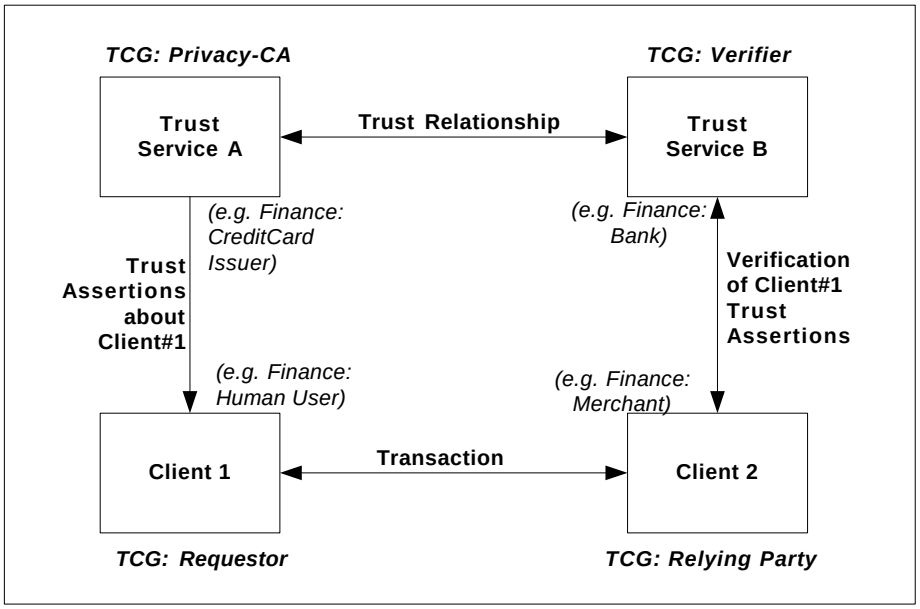


Figure 8: The classic Four Corners Model

The Four Corners Model can be best explained using an illustration from the financial industry. In this model, two clients (e.g. a Human User and Merchant) are conducting a transaction using a method of trust (e.g. credit card). The User has been assigned with a credit card by an Issuer (e.g. User's bank) as a representation of trust the Issuer in the User. The level of trust is expressed in the spending-limit on the credit card, which is a function of the credit rating of the User. The credit card is in fact a form of *trust assertion* regarding the User, issued by the Bank.

When the Merchant is presented with the User's credit card, the Merchant must verify the card-status of the credit card (i.e. still valid) and the credit-status of the card (e.g. credit available). In order to perform this verification, the Merchant must rely on another entity, typically a bank. In this sense, the Merchant is a *Relying Party* upon the bank (the *Verifier*).

In its turn, the Merchant's bank must perform the verification on the credit card, either through its own process (e.g. direct querying Visa, AMEX or MasterCard) or by querying the Issuer of the credit card. Since the financial industry is a tightly regulated industry and since all credit card issuers are regulated under the card brand (e.g. Visa, AMEX or MasterCard), there is a pre-existing *trust relationship* (direct or indirect) between the Merchant's bank and the Issuer of the User's credit card.

Note that both Trust Service A and Trust Service B in the Figure can be expressed or implemented as a single entity. In the credit card example, the Merchant's bank can be the same as the User's credit card Issuer. Thus, for all transactions with the User's credit card, the Merchant is verifying the card-status to the same entity as the card Issuer.

In the context of Trusted Platforms, the Basic Model of Figure 7 is a simplification of the classic Four Corner Model of Figure 8 where platform authentication of the Requestor's platform (Client 1) is done indirectly through the Relying Party (Client 2). Here, the Relying Party passes the assertions from (about) the Requestor to the Verifier, instead of the Requestor dealing directly with the Verifier. In some use case scenarios, the Verifier and the Privacy-CA can be implemented as a single entity.

4.5 Detailed Architecture for Deployment

Core to the value proposition of trusted computing is the ability of an entity (the *Requestor*) to provide security-related assertions or *attestations* regarding its platform to a second entity (the *Verifier*), who is authenticating the first entity (Figure 9). Thus, the Verifier is said to perform *platform authentication* of the Requestor, based on some *integrity measurements* and *integrity reporting* of the Requestor's platform which is presumed to have some protected capabilities.

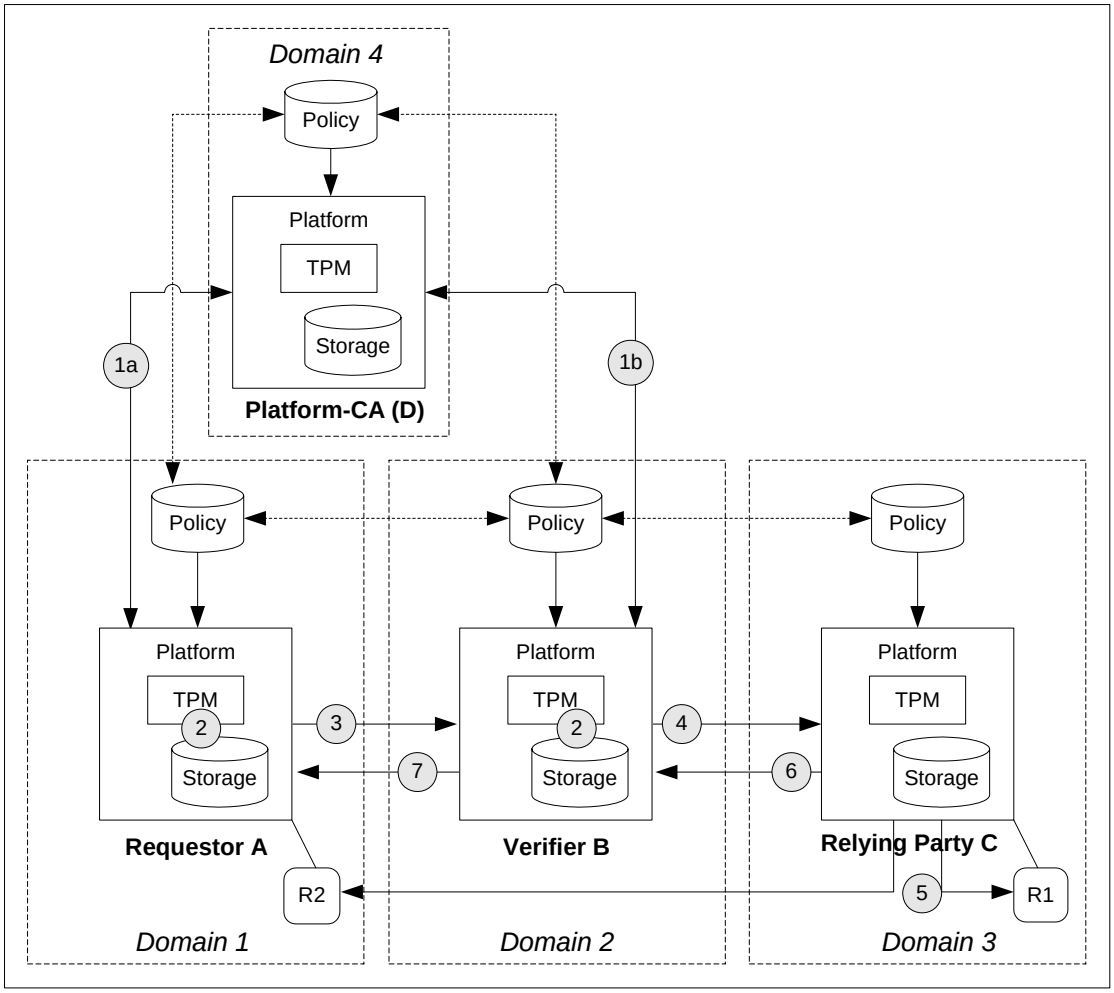


Figure 9: Detailed TP Deployment Architecture

In the process of performing platform authentication, the Verifier is evaluating attestations regarding the Requestor's platform. Although there are several forms of attestations in trusted computing, of particular interest here is the *attestation-of-the-platform* which can be briefly summarized as the operation that provides proof regarding the set of integrity measurements of a given platform. One way for a Requestor to provide attestation-of-the-platform (regarding its platform) to the Verifier is for the Requestor's platform to provide a set of its PCRs, signed using the Identity-Credential (AIK) found in its TPM.

Note that although the above Figure contains multiple domain boundaries, this does not preclude the scenario where all the entities reside within a single domain (e.g. Enterprise scenario). Here each entity is shown to be built using a trusted platform, containing a TPM and storage capacity for TPM-related objects (sealed and/or encrypted), including the Stored Measurement Log (SML),

keys and other objects. In addition, a policy object is shown representing the fact that policies govern the behaviors of each entity in the architecture and that some method of communicating policy information must be established among the entities.

4.6 Abstract Entities

Figure 9 illustrates the process of platform authentication in more detail. The entities shown are functional, and thus their roles may be interchanged depending on the use case scenario. The entities involved in the process of platform authentication are as follows:

4.6.1 Requestor

The Requestor is the platform seeking to be authenticated by the Verifier. Here the Requestor is assumed to be a trusted platform, possessing protected capabilities, integrity measurement functions and integrity reporting functions.

Note that in the case of *mutual platform authentication*, the roles of the Requestor and Verifier may be reversed.

4.6.2 Verifier

The Verifier is the entity who evaluates the assertions issued by the Requestor regarding the Requestor's platform. That is, for case of the attestation-of-the-platform, it is the Verifier that evaluates the attestations.

The Verifier function is distinguished from the Relying Party because is some areas application these are indeed separate physical entities. However, this does not preclude the oft-occurring cases where the Verifier is the consumer of its own evaluation results regarding the Requestor's platform.

The Verifier may be implemented also using a trusted platform. Note that in the case of *mutual platform authentication*, the roles of the Requestor and Verifier may be reversed.

4.6.3 Relying Party

The Relying Party is the entity that is dependent on the Verifier for evaluating the assertions regarding the Requestor's platform, using the attestation-of-the-platform approach.

The Relying Party itself may or may not be implemented using trusted platforms. Regardless, in the context of a given transaction or interaction with the Requestor, the Relying Party accepts the evaluation of the Verifier as being correct and sufficient according so some pre-agreed policy. Evaluations result can be binary or it can be a parameter within a given range (where the semantics of the range is understood by both the Verifier and the Relying Party).

Depending on the type of transaction upon which the Relying Party depends on the Verifier, the Relying Party may return a *response* to the Requestor, or it may perform some action which is meaningful to (to the Requestor) in the context of that transaction or interaction.

Note that the Relying Party and the Verifier is assumed to have performed their own authentication (one-way or mutual) prior to the Requestor seeking authentication. If the Relying Party is based on a trusted platform, then the same platform authentication process described here can (should) be deployed for these two entities.

Figure 9 shows two Relying Parties, C and D. C is a generic relying party while D is intended to be a Platform-CA that may exhibit the following properties:

- Multiple Instantiation - Distinct entities (in separate domains) may each implement a Platform-CA
- Role Distinction - Each of the Requestor, the Verifier and the Relying Party may employ Platform-CA functions within their domain according to domain policies.

Note that the Platform-CA in a domain could be a local (private) CA operated by the local IT administrator in that domain and whose identity credentials are relevant only in that domain.

4.6.4 Entities Encountered in the Deployment Lifecycle

There are likely to be several entities encountered at each phase of the deployment lifecycle. Entities logically satisfy one or more of the abstract entities described above. This section details specific entities anticipated in each phase of the lifecycle.

4.6.4.1 Manufacturing Phase

The primary entities involved in the Manufacturing Phase are as follows:

- Trust Credential Issuer – Attests to the association of an EK public key with a TPM / platform. (See 6).
- Integrity Values Provider – Computes integrity hash of components and makes them available for later comparison.
- Classic-CA – Issues X.509 or other certificate type that certifies a manufacturer's signing keys (See Section 6.5).
- Manufacturer – Fabricates, assembles and configures platform components, firmware and software. Manufacturers can be any participant in the supply chain.

4.6.4.2 Platform Delivery Phase

- Conformance Entity – Verifies platform design and assigns quality ratings. Conformance entities may issue credentials stating the results of their evaluations.

4.6.4.3 Platform Deployment Phase

- Platform Owner – Prepares the platform for operation and takes responsibility for establishing initial setup and configuration – may issue trust credentials and provide integrity values.
- DAA Issuer – Initializes DAA capability in platform and issues DAA credentials.

4.6.4.4 Platform Identity Registration Phase

- Platform-CA – Issues AIK credentials (See 7).
- DAA Verifier – Verifies the parameters obtained from the previous DAA-Issuer in the DAA-Sign protocol within a given use case scenario.
- Platform Owner – Registers platform identity with asset management database.
- Platform Operator – Configures customizes and personalizes the platform computing environment.
- Platform User – Enrolls for User credentials to be used on a given platform.

4.6.4.5 Platform Operation Phase

- Access Requestor – Seeks access to network resources (See 8.3).
- Policy Enforcement Point (PEP) – Controls access to a network (See 8.3).

- Policy Decision Point – Makes access control decisions (See 8.3 & 8.2).
- Remediation Service – Repairs configuration state or other unacceptable conditions.
- Certificate Authority – Reports revocation information.
- Migration Authority – Provides backup and migration of keys and data (See 8.4).

4.6.4.6 Platform Recycling and Retirement Phase

- Platform Owner – Prepares platform for decommission and change of ownership.

4.7 Platform Authentication Flows

There are a number of interactions among the entities in Figure 9 that must occur in order to achieve the platform authentication of the Requestor by the Verifier. These are expressed as flows in the figure, and are described further in the following.

- *Identity Credential Enrollment* (Flow 1a): Before a trusted platform (Requestor) can communicate meaningfully to the external world, it must first obtain an identity credential (e.g. AIK-certificate) from the Platform-CA.

  The process of identity-credential enrolment/registration and issuance from the Platform-CA requires that an entity possess the other TCG credentials (e.g. EK-credentials) prior to requesting the identity credential from the Platform-CA.

  The credentials profile and format are being address in IWG Credentials Profile document [2], while the process of enrollment has been address in the TLS context in the IWG TLS-Attestations Extensions document [6].

- *Identity Credential Publish* (Flow 1b): When a trusted platform (Verifier) wishes to verify the trustworthiness of another platform (Requestor), it must be able to obtain copies of the Identity Credential (AIK-certificates) of that second platform from one or more Platform-CAs.

  Note that this does not imply that both entities need to use the same Platform-CA to issue their respective identity-credentials. Rather, the intent is to convey the notion of *Register/Publish* behavior, in which the Verifier should have access to the Requestor's identity-credential either directly from the Requestor itself or indirectly from the Platform-CA whom issued the Requestor's identity-credential.

- *Platform Integrity Measurement* (Flow 2): Integrity measurement on a platform is the well defined process of obtaining metrics of the platform's characteristics which affect the integrity or trustworthiness of that platform. These metrics are stored in logs (Stored Measurement Logs), and digests (hashes) of them are put into PCRs.

  The platform integrity measurements of a Requestor platform are of primary importance to the Verifier in its evaluation the trustworthiness of the Requestor's platform. The Relying Party in ordinary circumstances is not interest in the details of the integrity measurement of the Requestor, and it relies on the Verifier to summarize these details into an evaluation result presented to the Relying Party.

- *Platform Integrity Reporting* (Flow 3): Integrity measurements regarding the Requestor's platform must be reported to the Verifier for evaluation by the Verifier.

  Two important aspects of this flow are the *structure and format* of the measurement logs communicated from the Requestor to the Verifier, and the transport *protocol* used to convey the measurements.

  The document [4] addresses the first need, while document [6] provides a way to communicate these measurements within the TLS protocol using extensions to the protocol.

- *Evaluation Reporting* (Flow 4): The evaluation result by the Verifier (of the Requestor) must be communicated to the Relying Party in a way meaningful to the Relying Party. This implies that some agreement has been reached between the Relying Party and the Verifier regarding the *evaluation metric* to be used by the Verifier. The precise metric of evaluation agreed to by the Verifier and the Relying Party is dependent on the context or use-case of the authentication based on trusted platforms.

  For example, the evaluation metrics used in a VPN scenario (e.g. where the Relying Party is a VPN Gateway) will be different from an Online Banking scenario (e.g. where the Relying Party is a payment processing system).

- *Direct Response/Action* (Flow 5): For a session or transaction within a given context employing trusted platform, the Relying Party may generate a Response and/or Action to the evaluation (of the Requestor) by the Verifier.

  The Response/Action (R1) may be local to the Relying Party in its domain and affects its domain only, or it may be a Response/Action (R2) that affects the Requestor in its domain. The model caters for both intra-domain and inter-domain responses and actions.

  An example of an inter-domain Response/Action would be the 802.1X authentication for Clients seeking authentication by an Authentication Server (AS) within an Enterprise domain. The Relying Party here is the Authenticator (802.11 Access Point), which is dependent on the evaluation of the Client by the AS.

- *Indirect Response/Action* (Flow 6 and 7): When a Requestor seeks a service from the Relying Party, the later may respond indirectly through the Verifier since the Verifier is both the Policy Decision Point and the mediator in communications between the Requestor and Relying Party.

  For example, in responses, such as success/fail (or authorize/unauthorized) can be communicated by the Relying Party to the Requestor through the Verifier. The Verifier may chose to reformat or interpret the response emanating from the Relying Party.

5 Entities, Assertions and Signed Structures

In this section we provide a discussion on the concept of assertions and trust in the context of the TP Lifecycle, as shown in Figure 3. The discussion is arranged around the notion of the entities produce information contributing to assertions, entities that produce authoritative assertions, the kinds of assertions that are relevant in trusted computing, and the possible embodiments of assertions in the TPMv1.1b and TPMv1.2 environments.

The purpose of enumerating the assertions is to clarify both the explicit and implicit trust assumptions and implications in entities performing actions in the TP Lifecycle. Thus, for example, a TPM hardware that does not have any EK Private Key instantiated (within the TPM) is indistinguishable from other TPM hardware from the same manufacturer. However, as soon an EK Private Key is made present inside that TPM, the manufacturers is essentially making the assertion A1 ("TPM contains a unique EK"). Note, however, that if another entity (other than the TPM manufacturer) made that EK Private Key present in the TPM (such as in the case of Post-Manufacturing or Late EK key generation), that entity still cannot claim the assertion A1.

Having a common list of assertions aids different entities in the TP Lifecycle to understand the security and trust implications of their behavior, and for a consumer of the technology to also understand what kind of Trusted Platform she or he is using in a given scenario.

Finally, the common list of assertions allows their corresponding labels to be used inside Integrity Assertions (e.g. Security Qualities field inside an EK Credential), thereby helping the consumer of that information (e.g. Privacy-CAs) to perform evaluations and issue further assertions that can be understood by other entities in the TP Lifecycle.

5.1 Entities producing assertions and signing them

The entities producing assertions are as follows:

Entity name	***Entity Label***
TPM Manufacturer (ISA component manufacturer)	Ent-1
Component manufacturer	Ent-2
TBB manufacturers (ISA component manufacturer)	Ent-3
RTM manufacturer (ISA component manufacturer)	Ent-4
Integrators (e.g. VARs, OEMs)	Ent-5
Evaluation laboratories[1]	Ent-6
Owner	Ent-7
Platform-CA	Ent-8
Verifier	Ent-9
DAA Issuer	Ent-10
Conformance Labs[2]	Ent-11

[1] The Evaluation Laboratory evaluates against some criteria (e.g. Common Criteria, FIPS).

[2] The Conformance Laboratory evaluates a product against the TCG specifications.

5.2 Types of assertions – What is signed

In the following, a classification of assertions is provided in the context of the phases within the TP Lifecycle diagram (see above).

5.2.1 TPM Manufacturer Assertions (Phase 1)

The assertions produced in this phase are as follows:

No.	Assertions meaning	Entity generating assertions
A1	TPM contains a unique EK.	**Ent-1**
A2	TPM EK is generated or injected.	**Ent-1**
A3	TPM correctly implements interfaces to a particular TCG TPM specifications.	**Ent-1, Ent-11**
A4	Common Criteria – meets referenced security evaluation standard).	**Ent-6**
A5	FIPS 140-2 Level X.	**Ent-6**
A6	Identify TPM Manufacturer, Model and Version at Manufacturing time.	**Ent-1**
A7	Correctly implements all semantics of the TCG TPM specifications.	**Ent-1, Ent-6**
A8	ISO900X certified.	**Ent-11**
A9	This is the EK public key in the TPM.	**Ent-1**

5.2.2 Platform Manufacturer Assertions (Phase 2)

The assertions produced in this phase are as follows:

No.	Assertions meaning	Entity generating assertions
B1	Conforms to a reference security target, protection profile and the elements of the assurance level.	**Ent-6**
B2	The platform contains a unique TPM	
B3	Identify Platform Manufacturer, Model and Version.	**Ent-5**
B4	Describes TPM physical binding mechanism	**Ent-3**
B5	Identify type of RTM	**Ent-4**
B6	Identify set of components included in RTM	**Ent-4, Ent-5**
B7	Identify other components of the platform	**Ent-5**
B8	Identify physical interface for the TPM	**Ent-5**

	connect	
B9	TCG Platform Specific Type/ specification version	**Ent-5**
B10	FIPS 140.2	**Ent-6**
B11	ISO900X certified.	**Ent-11**

5.2.3 Platform Delivery Assertions (Phase 3)

The assertions produced in this phase are as follows:

No.	Assertions meaning	Entity generating assertions
C1	ISO900X certified	**Ent-11**
	Assertions A1 to A7 inclusive.	
	Assertions B1 to B11 inclusive.	

5.2.4 Platform Deployment Assertions (Phase 4)

The assertions produced in this phase are as follows:

No.	Assertions meaning	Entity generating assertions
D1	ISO900X certified	**Ent-11**
	Assertions A1 to A8 inclusive.	
	Assertions B1 to B11 inclusive.	
	Assertions C1 to C3 inclusive.	

5.2.5 Platform Identity Registration (Phase 5)

The assertions produced in this phase are as follows:

No.	Assertions meaning	Entity generating assertions
E1	ISO900X certified	**Ent-11**
	Assertions A1 to A8 inclusive. (Excluding A9)	
	Assertions B1 to B11 inclusive.	
	Assertions C1 to C3 inclusive.	
	Assertion D1	

5.3 Trust Scores

At any stage of manufacturing, deployment and operation the entity generating assertions may also evaluate assertions made by other entities. Consumers of assertions trust the evaluation procedures of entities in the earlier stages. The most authoritative entity to make a trust score assertion is subjectively the evaluator of assertions defined in 5.2.1 - 5.2.5.

Often evaluations will produce acceptable results, but may vary in degree of confidence. It is believed that both discrete and aggregate confidence values are effective means to concisely capture non-binary evaluation results. *Trust Score* refers to an assertion of confidence regarding other assertions made by the entity (See 5.2.1 - 5.2.5).

Trust scores may be expressed in varying degrees of granularity. Therefore, trust scores should also be accompanied by a score basis. The score to basis ratio determines the relative confidence level the signer assigns to assertions it makes.

Potentially every assertion made can have a different confidence value. An amalgamation of confidence values may be made resulting in an overall composite trust score. The composite trust score is a required element that trivially has maximum confidence.

5.4 Impact of Credential Revocation on Assertions

The list of assertions above play an important role in the context of changes to a platform's set of credentials during its operation phase.

More specifically, there are certain circumstances, such as Field Upgrades and an Owner perform the *RevokeTrust* command, which results in the need of the platform to obtain new credentials with a different set of assertions implied by the new credentials.

6 Types of Credentials in the TP Lifecycle

The trusted computing ecosystem employs a number of credentials as a form of attesting the security properties of trusted platforms. Some of the credential contains a public key, while some do not (the later being known in the X.509 world as *attribute certificates*). These are briefly described in the following. For a definitive profile and explanation of the TCG credentials, the reader is directed to the IWG 1.1b credentials profiles document [2].

In discussing the various types of credentials and Integrity Assertions, it is useful to use the term *Platform-CA* as the entity that issues Identity Credentials to a given platform. Regardless of the implementation, the Platform CA is the entity that consumes all the integrity-related data and the credentials established in the previous phases:

- *The Privacy-CA*: The Privacy-CA is the entity to whom a platform requests an Identity Credential, which is then in-turn used to interact with a Verifier in a given platform-authentication event. The Privacy-CA is the consumer of all integrity information in the previous phases, and must be provided with sufficient information regarding the platform (e.g. EK-Credential, Platform Credential, other integrity assertions) in order to arrive at a decision to issue an Identity Credential (i.e. AIK-Certificate). The Privacy-CA is trusted to protect PII-related information regarding the requesting platform. The TCG definition of the Privacy-CA can be found in [12].
- *The DAA-Issuer*: The aim of the DAA-protocol is similar to that of the Privacy-CA, namely to establish an identity for the platform without unintentionally revealing PII. In the DAA-Join protocol the platform obtains DAA parameters from a DAA-Issuer, which are used in the DAA-Sign protocol with a verifier. Thus, the DAA-Issuer is assured through the DAA-Join protocol that the platform contains an EK-key residing inside the TPM. Typically the DAA-Issuer is the entity who generates the EK key and manufactures the TPM. The TCG summary of the DAA protocols and functions can be found in [12].

The DAA protocols were developed originally to address some of the privacy issues with regards to the Privacy-CA model, since the Privacy-CA by definition was in possession of platform-identifying information (e.g. EK-Credential). However, since real-world applications that presume the existence of a DAA-Verifier are yet to emerge, an intermediate bridging solution would be to combine the use of the DAA protocols with a Platform-CA. For example, a DAA-Issuer could exchange its classic identity certificate with the Platform-CA, allowing the Platform-CA to check that a Requestor entity (e.g. end-user) truly belongs to the group defined by the DAA-Issuer. Furthermore, the Platform-CA could issue an AIK-Credential to a Requestor only on the basis that the Requestor can prove possession of parameters provided by the DAA-Issuer.

This combined use of the DAA-protocols with a Privacy-CA leads to an interesting possibility in which the Verifier and the Relying Party (as defined by the Basic Model for authentication in Section 4.1) could in fact be implemented by the one Privacy-CA entity, thereby providing a practical use of the DAA concept. However, this notion needs further investigation as the DAA protocols are still under further refinement.

6.1 Endorsement (EK) Credentials

A given TPM is associated with an Endorsement Key, which is an RSA key pair and a certificate containing the public half of the key pair. The private half of this key-pair is held inside the TPM and is never revealed and is never accessible outside the TPM. Furthermore, the key-pair itself is never used to encrypt or decrypt user data.

A TPM can be recognized as a genuine TPM by asking it to prove its possession of an EK private key. This can be done by asking the TPM to decrypt information which is encrypted using the EK public key.

The primary consumer of an EK Credential is a Platform-CA. The EK Credential contains the Public Endorsement Key (PUBEK) which is used to unambiguously associate an AIK with a TPM

and platform. The EK Credential also contains information that may help an AIK issuer establish credibility in credentials it issues.

6.2 Platform Endorsement Credential

A *Platform Endorsement Credential* (or "Platform Credential" for short) typically a digital certificate, attests that a specific platform contains a unique TPM and TBB.

A *Trusted Building Block* (TBB) is the parts of the Root of Trust that do not have shielded locations or protected capabilities. Normally, this includes just the instructions for the RTM and the TPM initialization functions. The definition of a TBB is typically platform-specific. One example of a TBB – specific to the PC Client platform -- is the combination of the CRTM, connection of the CRTM storage to a motherboard, the connection of the TPM to a motherboard, and mechanisms for determining Physical Presence; for more information, see the TCG v1.2 PC Client Implementation Specification.

In general, the issuer of a Platform Credential is the platform manufacturer (e.g. OEM) and the consumer of the Platform Credential is a Platform-CA.

The Platform Credential contains information that the Platform-CA must use in attesting to the integrity characteristics of a platform. The Platform-CA may copy field-entries from the Platform Endorsement Credential to a new Identity (AIK) Credential that the Platform-CA creates for a Trusted Platform. An entity should not generate a Platform Endorsement Credential unless the entity is satisfied that the platform contains the TPM referenced inside the Platform Credential.

6.3 Attestation Identity (AIK) Credential

An AIK Credential is a certificate containing the public half of the RSA key pair that is associated with a given platform. The issuer of the AIK Credential is the entity referred to as the Platform-CA. In essence, the Platform-CA binds the AIK public key pair to a given trusted platform. This binding is expressed by the Platform-CA issuing (signing) a certificate containing the AIK public key and other attributes. More specifically, An AIK Credential contains the public portion of an AIK key generated by a TPM Owner, or TPM Owner delegate, that the Platform-CA gets in the AIK Credential request message from the TPM Owner, or TPM Owner delegate. The meaning and significance of the fields in an AIK Credential, and the Platform-CA signature over the fields in an AIK Credential, is a matter of policy; in general, though, the AIK Credential asserts that the public AIK contained in the AIK Credential is associated with a valid TPM.

For a given AIK key pair tied to a platform, the main purpose of the AIK private-key is to sign the PCR values which are then verifiable by anyone possessing a copy of the AIK public key. However, additional uses of the AIK-credential is allowed (e.g. see SKAE document).

Note that an AIK Credential can be issued in a "closed" domain by an entity in that domain, with the verifiers also within that same domain. Thus, for example, an IT Administrator within an Enterprise could issue AIK Credentials for all trusted platforms within that Enterprise. In essence, the IT Administrator takes-on the role of the Platform-CA. Unless other platform in other domains trusts that IT Administrator (as the Platform-CA in its domain), cross-domain interactions and transaction may not be trustworthy.

An Attestation Identity Key (AIK) is a special-purpose signature key created by the TPM; the AIK is an asymmetric key, the private portion of which is non-migratable and protected by the TPM. The public portion of an AIK is part of an AIK Credential request received by a Platform-CA and is also part of the AIK Credential issued by the Platform-CA. An AIK can only be created by the TPM Owner or a delegate authorized by the TPM Owner. The AIK can be used for platform authentication, platform attestation and certification of keys.

A TPM Owner is the entity responsible for the platform's security and privacy policies and is distinguished by knowledge of the Owner authorization data. The TPM Owner, or Owner delegate, sends an AIK Credential request to a Platform-CA; the request contains a public AIK, a TPM Endorsement (EK) Credential, and a Platform Endorsement Credential. The Platform-CA

may then produce an AIK Credential, after using the information in the request to verify the platform EK.

The AIK credential may be realized based on standard certificate formatting. A PKI specific profile may be required to map semantic differences that may exist given the context shift from user identity to platform identity.

6.4 Examples of Credentials in the TP Lifecycle

In order to understand more clearly the possible points within the TP Lifecycle where the various keys can be generated and their corresponding credentials issued, in the current section we discuss some examples using a basic Supply Chain diagram. The diagrams include the entities that are involved in the Supply Chain, the boundaries of the Lifecycle phases as well as the credentials that are issued.

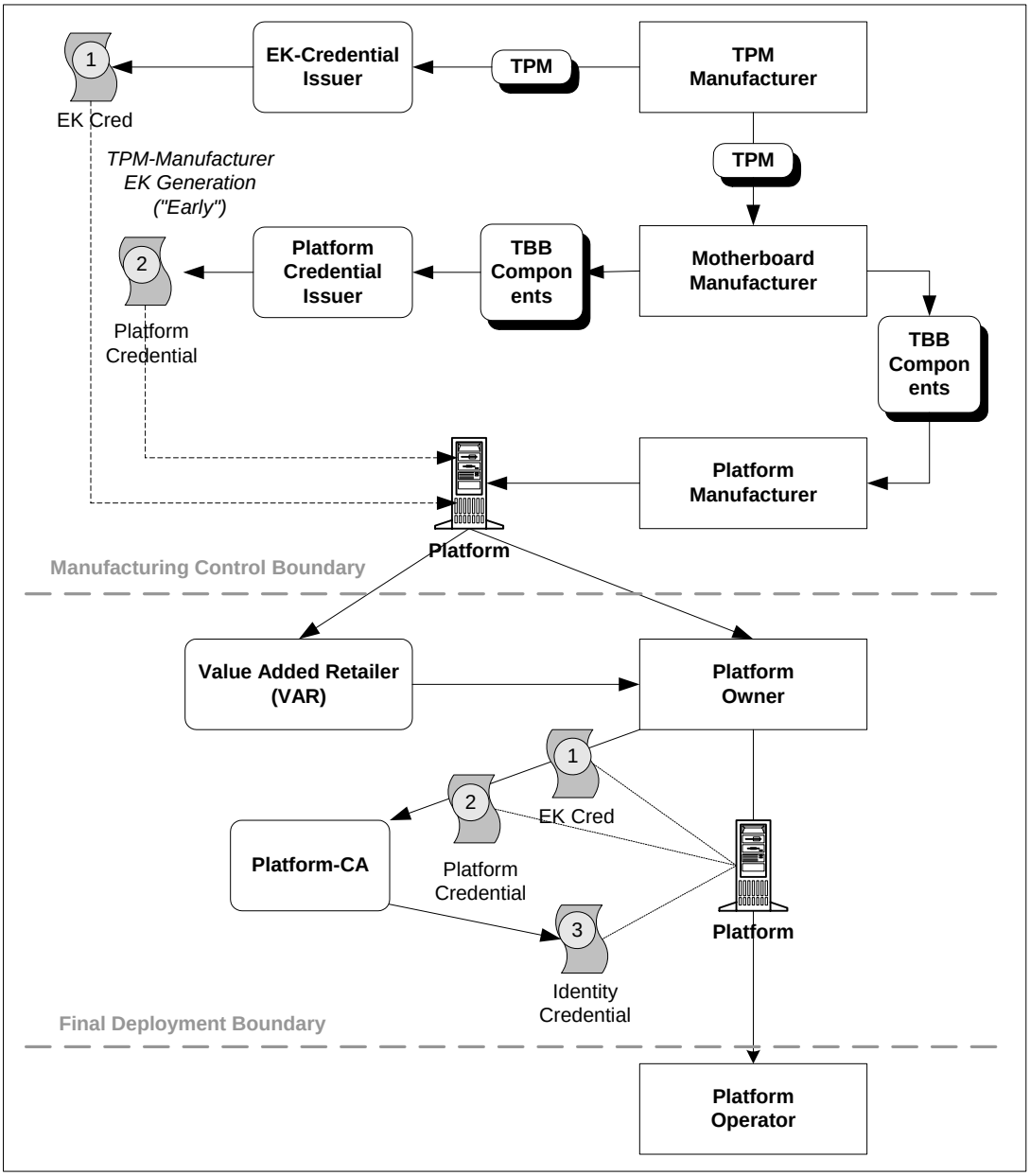


Figure 10: Example of Early EK Generation and EK-Credential Issuance

6.4.1 Example of Early EK-Credential Issuance

In normal circumstances, a TPM must contain an EK public key pair and EK-credential from the TPM manufacturer. This is referred to in the Lifecycle as TPM-Manufacturer Issued EK-Credential, or informally as “Early” EK public key pair and EK-Credential issuance. Figure 10 illustrates this case.

Early generation is the normative behavior because the TPM manufacturer is expected to be an organization, company or institution that is well recognized in the industry and has sufficient financial investments and legal obligation in being a TPM hardware manufacturer.

In Figure 10 the TPM Manufacturer is shown to be the entity generating the EK key pair and issuing the EK-credential. The Platform Manufacturer is shown to appoint a Platform Conformance Laboratory to perform conformance verification and to issue the Platform Endorsement Credential.

6.4.2 Example Late EK-Credential Issuance

Although in the TPM-1.1b specifications the EK key pair would be present in the TPM prior to the Platform Manufacturer (e.g. OEM) delivering the platform, the TPM-1.2 Lifecycle admits the technical possibility that an EK key pair be made present inside a TPM *after* the TPM hardware leaves its TPM-manufacturers premises.

The term employed to describe this is OEM-Issued EK-Credential, or more informally as “Late” EK-Credential, referring to the TPM Manufacturing phase as the delineation in time.

Figure 11 shows an example of Late EK Generation that is made possible by the design principles of the v1.2 TCG Specification set, and its impact on TCG Credential generation. Note that the entity labeled “Platform Conformance Lab” tests random sample(s) of from a lot of platforms of the same the same platform manufacturer and model (as defined in the Platform Endorsement Credential).

As an illustration, the figure shows the VAR to be the entity requesting the Platform Conformance Laboratory to issue both the EK-Credential and Platform Endorsement Credential. Notable here is the fact that this event occurs after the TPM Manufacturing phase.

Note that although not shown in Figure 11, it is quite conceivable that that Platform Owner be the entity that requests a trusted third party (such as the Platform Conformance Lab) to perform the same functions as requested by the VAR.

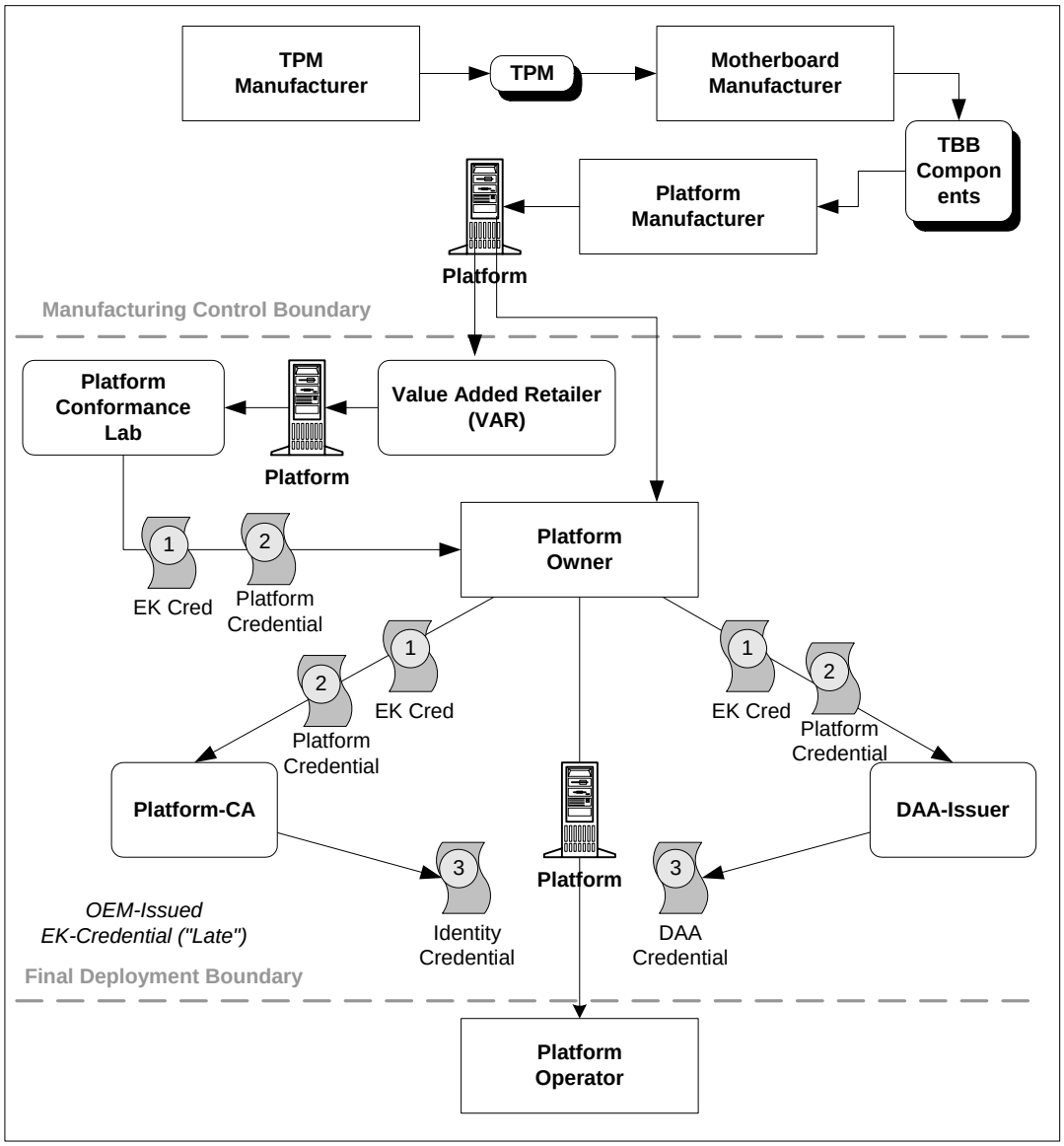


Figure 11: Example of Late EK Generation

6.5 Credential Management

Proper credentials management is crucial to the establishment of trust – both *social trust* and *technical trust* – in the credentials and in their issuing authority. Although many of the credentials that make-up a Trusted Platform maybe issued within a “clean room” environment by the same entities that manufacture the TPM and TBBs (or entities trusted by the manufacturers), other credentials may be issued and managed by entities outside the manufacturing boundary and positioned later in the TP lifecycle phases. Such entities may even be external to the deployment domain of a given TP.

In order to understand the appropriate credential management methods to be employed to manage credentials on a TP, it is useful to delineate the types of credentials and issuing authorities along the lines of the lifecycle infrastructures:

- *TP-Predeployment credential management*: Credential management here pertains to the EK-credential, and Validation credentials.
- *TP-Deployment credential management*: During deployment of a TP by an Owner, the relevant credentials include the AIK-credentials and user certificates (possibly derived from AIK-credentials).
- *TP-Recycling credential management*: When a TP is to be retired, then besides the *revocation* of the relevant credentials on the TP, there is also the issue of long-term *archiving*/warehousing of copies of credentials (e.g. AIK-credentials) that may be necessary for legal purposes (e.g. digital notarization use case).

6.5.1 Basic Credential Management Model

In the traditional or classic model for certificate management, two functions are identified on the side of the certificate issuance authority. The first is the Registration Authority (RA), while the second being the Certificate Authority (CA) as shown in Figure 12.

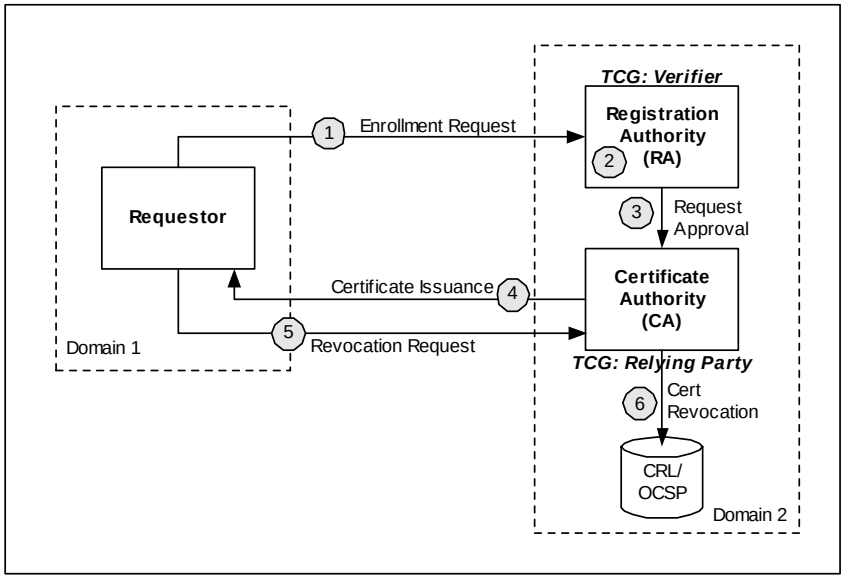


Figure 12: Basic Credential Management Model

In the classic certificate management model, the RA is treated as a separate function from the CA as it is the RA that evaluates the trustworthiness and reputation of the Requestor. For a given certificate enrollment request (Step-1), the CA relies on the outcome of the evaluation of the RA regarding the Requestor (Step-2). A positive decision from the RA (Step-3) results in the CA issuing the certificate to the Requestor (Step-4). When the Requestor asks the CA to revoke a given certificate (Step-5), the CA publishes the revoked certificate (i.e. serial number) using either the CRL mechanism or the OCSP mechanism (Step-6).

In practice, the RA and CA are implemented by the same entity or institution due to the close relationship between the two functions in the context of establishing a credential, namely a certificate, which is a form of expression of social trust in the digital world.

The credential management model of Figure 12 maps readily into the context of trusted platforms as the RA-entity will be the Verifier of the Requestor's platform, and will only approve credential issuance if the Requestor's platform conforms to that expected by the Verifier. In other words, the

credential management model is applicable to the Platform-CA whose task is to evaluate the Requestor's platform and issue AIK-Credentials to the platform.

6.5.2 Credential Management Protocols

The area of PKI and credentials management is relatively mature, with several credential management protocol having been proposed in the past few years. For the IWG, therefore, one possible avenue would be to develop an extension to one or more of the widely deployed protocols which can provide a way to convey platform-related information. Such an extension to the protocol should allow for both:

- A platform without an AIK-credential to obtain one (ore more) AIK-credentials from the Platform-CA
- A platform with an AIK-credential to obtain one (or more) classic certificates from a Classic-CA, based on the requestor's trusted platform and AIK-Credential.

In-line with the development of the credentials profile in [2], in which credentials are specific in both X.509 and XML format, two protocols for certificate management are the CMC protocol defined in RFC2797 (see [13]) for X.509 certificates, and the XKMS protocol (see [14]) for XML-based credentials.

- **X.509 certificates: *Certificate Management Messages over CMS* (CMC)**

  The CMC protocol is a product of the IETF PKIX working group in the effort to develop a simple yet functionally rich protocol for credential management. CMC uses the PKCS#10 Certificate Request Syntax standard for a basic request format, and the PKCS#7 Cryptographic Message Syntax for protecting exchanged messages.

  For a more complete functionally, CMC uses the Certificate Request Message Format (CRMF) as defined in RFC2511 (see [15]). For message encryption and signatures CMC uses the Cryptographic Message Syntax (CMS) as defined in RFC2630 (see [16]), which provide some improvements over the original PKCS#7.

- **XML certificates: *XML Key Management Services* (XKMS)**

  The XKMS protocol [14] is really a request-response protocol layered on SOAP. It provides a "binding of keys to entities, thereby providing some wrapping or abstraction above the actual PKI engine underlying a given implementation".

  XKMS can be viewed as consisting of two parts:

  - The XML Key Registration Service (XKRSS): The XKRSS supports four (4) services. These are: *register*, *recover*, *reissue* and *revoke*. Each of these services present a number of individual credential management functions. As a whole, the four services can provide a complete lifecycle support for credentials.
  - The XML Key Information Service Specification (XKISS): Credential look-up and validation is supported using XKISS, which consists of two services: *locate* and *validate*. Using these two services, two communicating parties can discover which credentials to employ within a transaction and obtain status-validation information regarding the credentials being deployed.

  XKMS provides a way to express certificate management function is XML, while providing a wrapper over legacy CA services designed for X.509 certificates. As such, XKMS provides the most attractive solution for credential management for existing CAs in the PKI industry. XKMS has completed standardization in the W3C.

6.5.3 Certificate Policy for TCG Credentials

For credentials to be interoperable across domains, in addition to syntax-related requirements there is the question of the semantics and intended use of the credentials. When a credential Issuing Authority (e.g. CA) signs/issues a certificate (to a given entity), it is essentially making a statement to the consumer (of that entity's certificate) that a particular public key is bound to a particular entity (the certificate subject). The extent to which that consumer should rely on this statement by the Issuing Authority needs to be assessed by the consumer. The intended use of a certificate is typically expressed in a *Certificate Policy*, as defined in RFC2527 (see [18]).

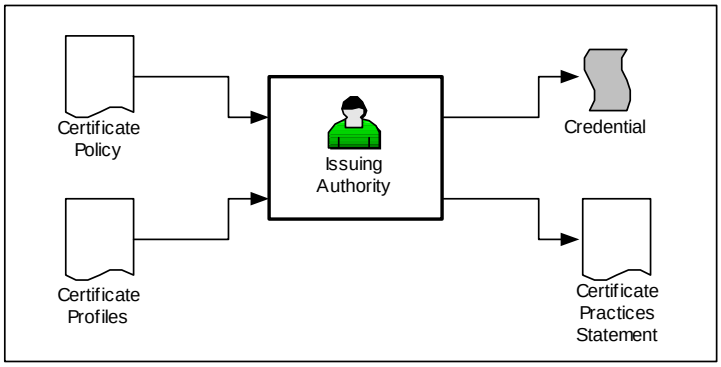


Figure 13: Certificates and CPS

RCF2527 defines a Certificate policy as a named set of rules that indicates the applicability of a certificate to a particular community and/or class of application with common security requirements. For example, a particular certificate policy might indicate applicability of a type of certificate to the authentication of electronic data interchange transactions for the trading of goods within a given price range. Thus, a certificate policy provides an explanation of the context within which a given certificate was issued and the intended use of the certificate.

In X.509 certificates, the certificate policies extension indicates the policies under which the certificate was issued. When used within a CA certificate (e.g. root certificate), it indicates the policies under which the CA operates. When found within end entity (e.g. user, device) certificates, it indicates the policies under which the certificate was issued.

In the context of the TCG credentials, a certificate policy must be developed by the TCG for all the credentials in order explain the context within which a given TCG-credential should be issued and the intended use of the credential. Thus, for example, the TCG Certificate Policy should clearly indicate that an EK Credential is accessible only to functions within a trusted platform and cannot be used to sign external data objects, and so on.

7 Privacy Issues

Privacy in the digital world of today is an important matter, and in many environments the acceptability of new technologies is dependent on the ability of the technology to preserve the privacy of the user and his/her platform by way of hiding or removing Privacy Identifying Information (PII) in transactions and interactions with the platform.

Within the context of TCG technology the identity of the platform is typically represented by the Attestation Identity Key (AIK) credential, which is issued by the Privacy Certificate Authority (Privacy-CA). The term "privacy" in Privacy-CA is intended to mean "privacy-preserving" and is used to denote the fact that the entity must be trusted to guard PII-related information from user or entities other than the Owner of the platform (and Users of the platform authorized by the Owner). For example, when a platform seeks to obtain an AIK-credential from a Privacy-CA, it must sign its request using the EK private-key (of its TPM) and deliver a copy of the EK-credential to the Privacy-CA. This allows the Privacy-CA to in fact make a correlation between the AIK-credential (that it issues to the platform) and the physical platform itself through the EK-credential of the platform. Hence the acute need of the Privacy-CA to maintain the privacy of this correlation information.

Although it is common business practice today for many CAs to maintain as private the business information and user information of the entities to whom the CA issues certificates, within the context of TCG technology there is a concern that in certain circumstances no entity (not even a public legal CA) can be trusted to maintain PII-related information regarding a platform. In these circumstances a different approach must be adopted in the way of using an anonymous method to obtain an identity-credential for the platform.

It is to address this specific and stringent need of privacy at the TPM level and platform level that the TCG has developed the *Direct Anonymous Attestations* (DAA) protocol. The aim of the DAA protocol in simple terms is to provide a method for a platform to obtain an anonymous platform credential such that its issuer or end-consumer (i.e. Verifier) cannot make a correlation to the possessor of the anonymous platform credential. Thus, in DAA there is no CA entity that is trusted with the EK public key of the platform.

It is important to realize that obtaining absolute privacy on the open Internet at the user level may be prohibitive in cost, and therefore unrealistic to achieve. Thus, even if a user on a trusted platform employs an anonymous platform credential, other correlative methods may still be used at the application layer (e.g. browser) which tracks the user's behavior, regardless of the underlying platform that the user employs. Thus, it is crucial for both the Owner and the User to understand the purpose and scope of anonymous platform credentials in the TCG, notably the DAA protocol.

These two approaches – using a Privacy-CA and using the DAA protocol – are discussed in the following.

7.1 Role of the Platform-CA / Privacy-CA

The overall role of the Platform-CA is to vouch to the external world that a given platform is truly a trusted platform as defined by the TCG, and issue an AIK-Credential to state that fact. In contrast, the role of a Classic-CA is to attest to the world that a given individual or company has been assigned a certain public key pair. Thus, although the two entities may use the same mechanisms to achieve similar aims, there are some underlying differences between the two that needs to be emphasized.

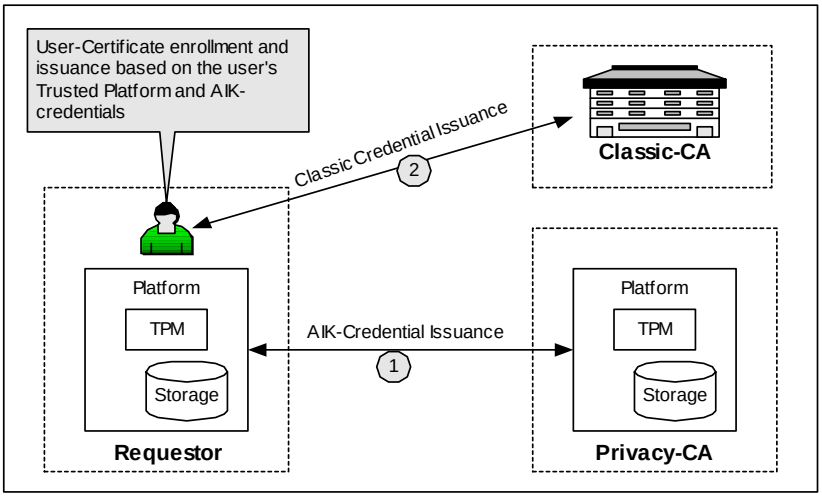


Figure 14: Classic-cert issuance based on TP and AIK-Credentials

The Privacy-CA is a Platform-CA that is also trusted not to disclose EK public keys of registrants using traditional AIK registration protocols. Although a Classic-CA may take-on the role of the Privacy-CA, within the trusted computing overall design and architecture the two roles have been made distinct both for clarity in problem definition and to reflect the notion of privacy or "blinding" (of a platform's true identity) that needs to be performed by the Privacy-CA.

From the perspective of a Classic-CA, a trusted platform provides benefits of better security for the transactions involved in credential management. Thus, for example, a user could use an AIK key pair for encrypting and/or signing certificate request messages sent to the Classic-CA (Figure 14). The Classic-CA then delivers the user's certificate encrypted using the AIK public key, guaranteeing that the message is decipherable only on the same platform on which the user initially issued the request. Similarly, the revocation mechanisms used by a Classic-CA can be used to also propagate information regarding revoked AIK-credentials issued by a Privacy-CA, who may be a different entity from the Classic-CA.

It is possible for there to be multiple Platform-CAs specific to Requestor, Verifier and Relying Party respectively. This suggests interoperability requirements with regards to the Identity Credentials (AIK-certs) issued by Platform-CAs and the need for a common set of certificate practices statements (CPS), based on a common certificate practices framework (see RFC2527).

7.2 DAA Protocols

In a number of use cases, the privacy of the user deploying a platform can be of utmost importance, such that even the Platform-CA is not trusted by the user. For such use case, an alternative to using a Platform-CA is to deploy the Direct Anonymous Attestations (DAA) protocol to obtain DAA-Credentials.

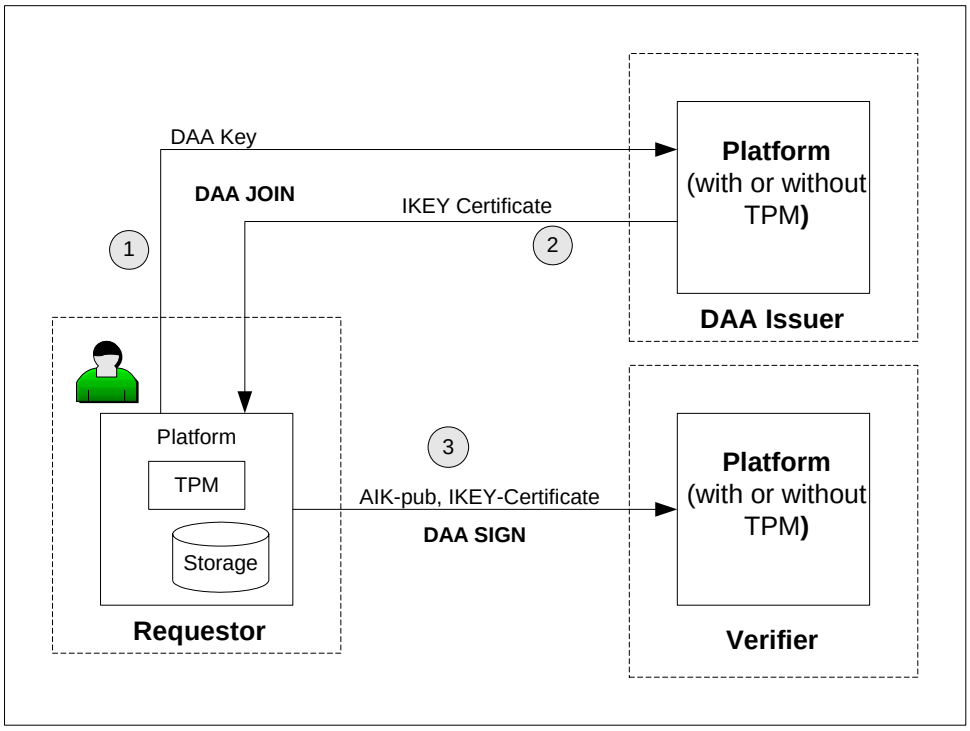


Figure 15: The DAA Protocol

The DAA protocol consists of two sub-protocols. The first, called the DAA-Join protocol, the TPM/Host needs to obtain a DAA-Certificate from a DAA-Issuer. This certificate is then used in the second sub-protocol, namely the DAA-Sign protocol, to engage with a Verifier (see Figure 15). The advantage of the DAA approach is that it provides the highest level of anonymity of a Trusted Platform when the platform interacts with the external world, whilst still being able to prove that the platform is a TP as defined by the TCG.

The DAA scheme involves four entities: Issuer, TPM, Host, Verifier, and two protocols, namely the Join and Sign protocols. The Host is defined to be the platform that contains the TPM. The computation required to perform a DAA protocol is split between the Host and the TPM. Any computation that is not required for security reasons to be performed by the TPM is given to the Host to perform so that the protocol can be more efficient. As such, the Host is assumed to be a much faster processor than the TPM. Note that the Host is also assumed to be trusted to protect host-controlled PII and PI related information.

The DAA protocols can be informally described as follows.

- An Issuer generates a public key, called IKEY (or Issuer Key), and a corresponding private key.
- The Join protocol is a protocol between the Issuer and a (TPM, Host) pair, in which the (TPM, Host) pair receives a IKEY-Certificate. In the JOIN protocol, the TPM generates a private DAA key, called privDK. The TPM identifies itself to an Issuer using the Endorsement Key (EK) of the TPM. If the Join process is successful, the Issuer will provide a parameter called certDK to the TPM and the Host. With the pair privDK and certDK the TPM now has the ability to sign a message that can be verifier using IKEY-Certificate.

The DAA Join protocol results in the platform containing certDK values. However, only the IKEY is needed by verifiers. Logically, the Verifier receives a certificate so that it can verify challengers. In this model only the IKEY is communicated to the verifier and should be signed by the Issuer. We refer to this as the "IKEY-Certificate". The IKEY-Certificate is also communicated to the Host allowing the host to verify the IKEY Issuer identity. Thus, in reality the other "certDK" parameters are protocol elements that the Host maintains for use during DAA Sign, but do not need to be used by the TPM. They are not used by Verifiers.

The Sign protocol is a protocol between a Verifier and a (TPM, Host) pair, in which the Verifier gets assurance that an AIK is held by a valid TPM. In the Sign protocol, the TPM will generate an Attestation Identity Key (AIK). The TPM and the Host will give the AIK-public-key to the Verifier. The TPM and the Host will also use privDK and certDK to generate a signature on the AIK-public-key, and also provide that signature to the Verifier. The Verifier will then use the IKEY Certificate to verify that the signature is valid. If the signature is valid, then the Verifier will be convinced that the AIK-private-key is held by a TPM.

8 Specifications Roadmap

In order to seed the development of the ecosystem that supports the proper functioning of a Trusted Platform (TP) and to seed new functions and services on the Internet that make use of TP features, some fundamental functions and services must be made available to support the operational aspects of a TP. To this end, a number or specifications have been developed to support the deployment infrastructure for TPs. These are discussed in the following.

8.1 Credentials Profiles

The purpose of Credentials Profiles specification [2] is to collect, in one document, definitions for three of the credential types identified in the v1.1b TCG Main specification, namely, the TPM Endorsement (EK) Credential, the Identity (AIK) Credential, and the Platform Endorsement (Platform) Credential.

8.1.1 Credentials Profiles v1.1b and v1.2

For all three Credential types, this specification includes, at a minimum: (a) an x.509v3 example Credential, the wire format for existing PKI infrastructures; (b) an XML example Credential, the wire format for Web services; and (c) an IDL definition of the Credential fields for RPC wire format. The intended audience for this document is people who work for the entities, such as Platform-CAs, who are expected to participate in the TCG infrastructure.

People who work for computer OEMs and the companies in the OEM supply chain, such as TPM vendors and software vendors, are also intended audiences for this document.

The completeness of the Credential profiles specifications in this document will be judged using the following criteria:

- Interoperability
- Backward compatibility with Section 4.32, Credentials, and Section 9.5, Instantiation of Credentials as Certificates, in the Version 1.1b TCPA Main Specification, dated 22 February, 2002.
- Trusted Platform owner and user privacy protection
- Credential profiles and formats support the IWG Use Cases as well as the protocols described in the current document.
- Credential validity check and revocation features are appropriate to the credential type; for example, these features are optional for TPM Endorsement (EK) Credentials but are required for the other credential types. Note that the certificate structure standard, such as x509v3 or XML, used to instantiate the credential type, may require a validity check and/or revocation field even if it is optional for the credential type. For working definitions of the terms “credential” and “certificate,” see section 1.6. of the Credentials Profiles specification.

Note that the notion of Validation Credentials have been deprecated for v1.1 credentials due to the fact its need is not anticipated in the deployment of v1.1 TPMs.

8.1.2 DAA and IKEY Credentials

The purpose of the Direct Anonymous Attestation (DAA) protocol is to convince a verifier that an AIK (Attestation Identity Key) is held by a TPM without allowing multiple verifiers to corroborate transactions involving different AIKs from the same platform. Furthermore, DAA helps achieve the objective without requiring a trusted third party. Verifiers rely on DAA Issuers to establish a group from which the verifier cannot distinguish between other platforms. The DAA Issuer issues an IKEY certificate to communicate a public key that verifies group member’s private keys. The DAA Issuer can make assertions common to all platforms in the group. The works of [8] , [10], [11] and

[22] discuss the DAA and the properties that are achieved by DAA. The document assumes that the reader is somewhat familiar with the basic TPM (Trusted Platform Module) functionality that is defined by the Trusted Computing Group (TCG).

8.2 Managing Platform Integrity

An aim of trusted computing is managing platform configuration and changes to the computing environment. To accomplish this aim configuration state needs to be represented unambiguously and that state needs to be authenticated. Platform integrity information can be divided into two categories, integrity assertions and integrity values. Assertions are enumerated claims regarding intrinsic attributes of a platform or one of its components. Integrity values are metrics that unambiguously identify platform firmware and software, such as a message digest.

Integrity assertions and values are associated with a platform in a couple of ways.

- In-band collection and reporting - integrity values / assertions are intrinsically bound to the platform and a measurement agent calculates the measurement digest and report to a verifier.
- Out-of-band collection and reporting - integrity values / assertions are captured by manufacturing or deployment processes and distributed to verifiers.

Integrity attributes of a platform need to be authenticated prior to use. It is necessary to authenticate the platform such that the authentication can be linked to integrity attributes. The AIK is used to authenticate and AIK credential to establish the link.

8.2.1 Platform Integrity Information Schema

The specification [4] is concerned with integrity management infrastructure that touches manufacturers, verifiers and platform owner entities. This specification addresses interoperability as it relates to the production, collection, communication and evaluation of integrity information.

Architects, designers, developers and technologists who are interested in the development, deployment and interoperation of trusted systems may find this document helpful in providing both abstract and implementation specific insights for achieving interoperation between TCG-based systems.

The TCG integrity management model covers the production, collection, communication, storage and evaluation of integrity values related to platform configuration state. The collection of integrity values associated with both static and dynamic integrity state is contemplated. Interoperability is a primary concern that impacts:

- Producers of integrity values (manufacturers, OEMs, product vendors)
- Platform measurement and reporting agents
- Platform verification agents

Integrity information must be communicated over the Internet and arbitrary intranets. It must be in a form amenable to transport and session layer protocols. To achieve interoperability, measurement log structures and manufacturer produced integrity values should be based on a common format specification. A common API is not a goal of this specification.

Parts of this specification may move into a future Credential Profile for v1.2 specification. For example, Integrity information ties together the phases of platform lifecycle with assertions of trust (EK & Platform credentials), identity (AIK credential) and state (PCRs). It makes possible resolution of inferences regarding acceptable operational state that are preconditions of trusted computerized interaction.

8.2.2 Platform Authentication and Attestation Protocols

The *TLS Extensions for Attestation* (TLS-Attestation) Specification for TPMv1.2 specification (see [6]) defines an extension to the RFC2246, building upon RFC3546 which describes a uniform way to extend the TLS handshake protocol defined in RFC2246. [6] is an example of one implementation approach for platform authentication and attestation. It may be appropriate to extend other handshake protocols or invent new protocols. Protocol definition is an ongoing effort within the IWG and the TNC subgroup.

The TLS-Attestation specification defines extensions that would permit:

- platform authentication based on a TCG Attestation Identity Key (AIK),
- platform configuration reporting in the context of a platform authentication session, and
- platform registration / enrollment with a trusted service or host.

The TLS Extensions approach was selected because TLS (SSLv3) is widely used in the Internet today, and using TLS also addresses the growing body of EAP methods in 802.1X that build upon TLS (for example, EAP-TLS, EAP-TTLS and PEAP).

The basic TLS protocol permits the exchange of certificates for client and server authentication. Traditional certificates rely on an external mechanism for associating public keys with names. In practice, the TLS server uses a “machine certificate” which may include the DNS domain name, machine name and company name. A more robust solution will leverage TPM platform identifiers for both client and server identification based on AIK credentials.

TLS extensions are an IETF standard mechanism for extending TLS handshake exchanges. TLS attestation extends the TLS handshake to authenticate a platform independent of user or application authentication and to perform TCG platform identity registration. TCG extensions are symmetrical allowing peer-to-peer handshake semantics.

8.3 Trusted Network Connect

The Trusted Network Connect focus area is a specific use case of the IWG architecture that addresses the interest of network operators or administrators in enforcing policies regarding endpoint integrity when granting access to a network infrastructure. These policies often include the authentication of the endpoint user, as well as verifying that the endpoint hardware and software state meet established conditions. Example conditions may include establishing that certain software conditions are present (e.g. operating system version and patch level are current, that anti-virus software is present and operational, and that the anti-virus signature definition files are the most-recent version available). Hardware conditions might include policy requirements that the hardware be a Trusted Platform.

8.3.1 Network Authentication relationship to IWG Architecture

Established industry models exist documenting commonly-accepted architectures for network authentication based on user authentication. In the case of IP networks, the IEEE 802.1x authentication model, combined with IETF RADIUS and IETF EAP (RFC2284) standards outline an extensible architecture that has been broadly implemented in a variety of environments.

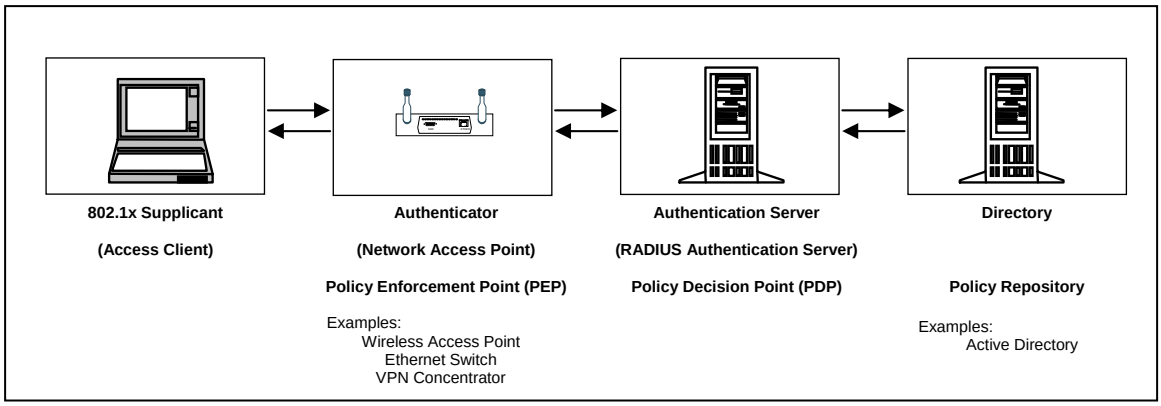


Figure 16: Traditional IEEE 802.1x / IETF RADIUS Network Authentication Model

Figure 16 outlines the four basic elements of the 802.1x / RADIUS network authentication model as it is typically described.

Figure 17 re-states these network authentication elements into the core components of the IWG architectural model.

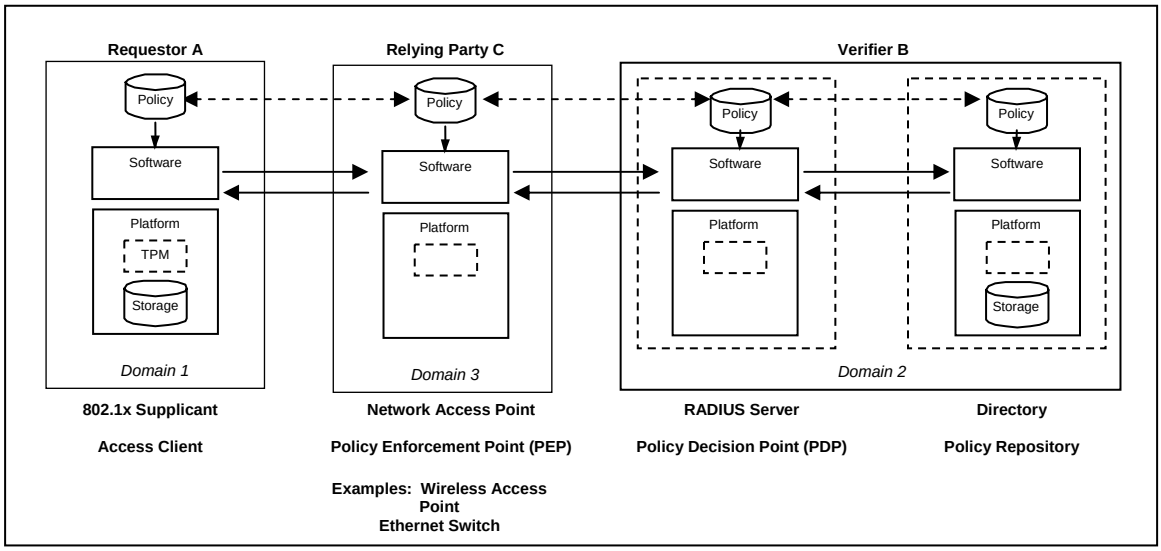


Figure 17: Network Authentication related to IWG Architecture

As can be seen in Figure 17, the 'Verifier' role defined in the IWG architecture is further sub-divided in the network authentication model into separate roles where the RADIUS server provides the authentication decision-making, while the policies and user-name/password directory information is managed in a separate Directory or repository.

In this architectural implementation, the role of the network access point is to accept requests for network authentication from clients (supplicants), and pass these requests for authentication to the authentication server (RADIUS). Depending on the response from the authentication server, the network access point will then enable the client to access selected areas of the network (either full access, access to selected VLANs, or no access). In the case where the authentication server determines that remediation might be required due to specific conditions on the client (requiring updated virus signature files, for example), the authentication server can direct the network access point to grant the requesting client access to a specific VLAN that may

have been pre-configured to allow only the specific remediation required (for example, downloading the necessary virus signature files).

8.3.2 Protocols Used in Network Authentication

There are a number of protocols used in network authentication in a traditional 802.1x / RADIUS authentication scheme. Figure 18 illustrates the protocols and the roles played by each of the actors in the authentication process.

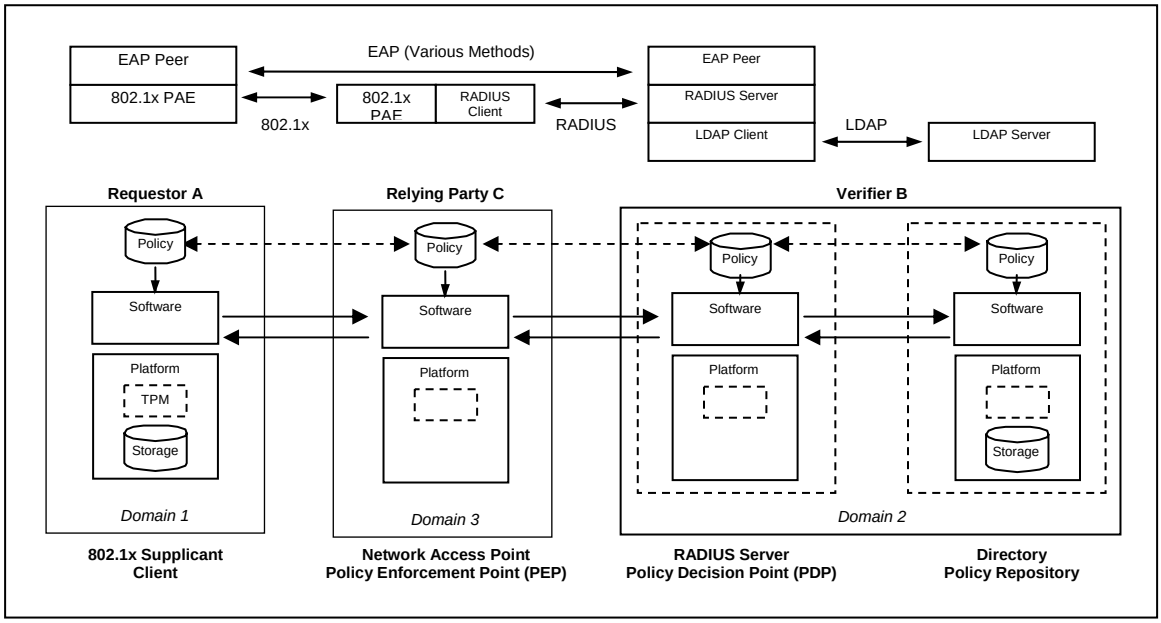


Figure 18 Protocols Used in Network Authentication

In the Trusted Network Connect use model, the traditional network authentication approach must be extended to include additional trust attributes in the authentication sequence. These additional elements would include attributes related to the state of the client platform and related trust credentials.

A key role is played by software on the client, extending the traditional 802.1x supplicant client process with the addition or insertion of these additional trust credentials into the authentication stream. This extended client software capability is referred to as the Trusted Network Connect client agent.

The role of the verifier must also be extended from the traditional network authentication approach to include the verification of these additional trust credentials. Figure 19 depicts the extensions to the network authentication architecture.

From Figure 19, it is apparent that the role of the Verifier in the Trusted Network Connect use case may actually be spread across multiple devices or elements. Detailed architectural options for structuring the elements required in the Trusted Network Connect Verifier function are described in more detail in architectural documents that will be delivered from the Trusted Network Connect subgroup.

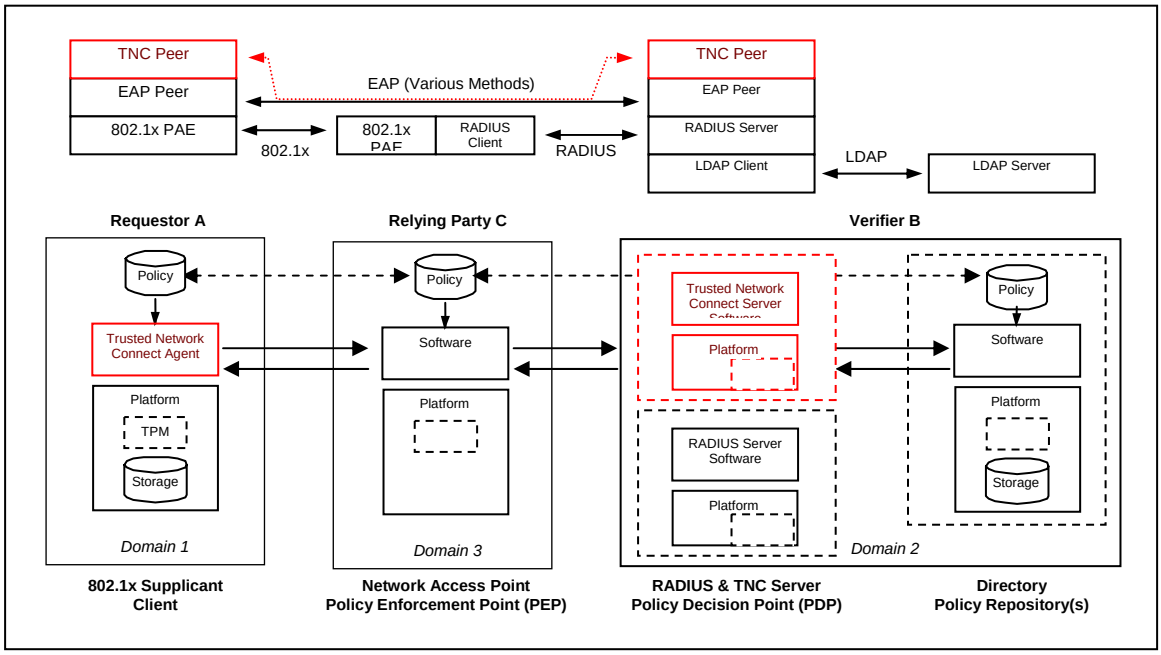


Figure 19 Trusted Network Connect extensions to the Network Authentication Architecture

8.4 Backup & Migration

The Backup-Migration specification is directed towards developing interoperable Key Backup and Key Migration Services for TCG platforms.

Migration is a TCG specific operation that allows for the secure movement of migratory cryptographic keys from one TCG compliant platform to another compliant platform in such a fashion as to allow the new environment to function in a similar manner, with respect to the usage of the cryptographic keys. These operations are effected with the use of a Migration Authority and a Migration Selection Authority.

This document is intended to serve as the living design document for the development of the Data Backup and Recovery specifications of the TCG Infrastructure Workgroup. It details the specification as well as provides a historical record of decisions that were made in developing this specification.

A v1.0 release of this document to an audience outside the TCG is targeted for 3Q04. The scope of the initial release may be limited to specific v1.1b relevant Use Cases, with follow on work to address a more comprehensive set of Use Cases including support for CMKs and other v1.2 dependencies.

The intended audience for this document is IT professionals who wish to develop or understand Migration Authorities and Migration Selection Authorities, or who have interest in key Migration and Backup operations as they apply to TCG compliant platforms.

Certain portions of this specification may be relevant to professionals who work for computer OEMs and the companies in the OEM supply chain, such as TPM vendors and software vendors, in order to ensure the development of interoperable products.

It is expected that the professionals attempting to comprehend this specification will have a working knowledge of TCG fundaments, especially regarding the TPM and TSS. Additionally, a

rudimentary understanding of Internet technologies, web services architecture and distributed computing concepts is highly desirable. Internet messaging technologies include HTTP, SOAP and TLS as well as an understanding of internet data description languages such as XML, XKMS, WSDL and ODRL will be of value to readers. An understanding of Public Key Infrastructure (PKI), certificate encoding technologies, XML Signatures and XML Encryption may also be beneficial.

This document is intended to provide:

- An interoperable reference specification for TCG clients and servers to interact in order to effect Migration services across a private or public network,
- Interoperable services for PC clients using either MSCAPI or PKCS#11 CSPs,
- Extensibility for other CSPs

8.5 Subject Key Attestation Evidence

One interesting use of an AIK-credential is to increase the assurance level when a user requests a certificate from a Classic-CA. The *Subject Key Attestation Evidence* (SKAE) specifications provides an extension and several possible scenarios to determine the usage of TCG compliant platform and TCG Identity keys that had been used to certify TPM signing and/or binding keys for X.509 public key infrastructure.

The SKAE extension conveys the certification evidence of the key referenced in the TPM_CERTIFY_INFO by the attested Identity Key (AIK) using TCG enabled platform. It could be used as an extension in X.509v.3 certificates as defined in RFC3280 (see [19]), attribute certificates as defined in RFC3281 (see [20]), certificate requests as found in RFC2511 (see [15]), various authentication and authorization protocols or elsewhere.

The value proposition that underlies this specification is that the trustworthiness of the TPM certified key can be attested by an AIK Credential. The introduction of the new extension provides a standard mechanism (see RFC3126 [21]) by which the binding of TPM certified key with an AIK can be verified. This feature would leverage the interoperability of TPM and legacy security systems since X.509v3 public key certificates are extensible and can be used for authentication and/or authorization, whereas the verification of this extension can be delegated to other TCG aware applications.

Note that an AIK can certify (cryptographically bind) only non-migratable or CMK keys.

Fundamental terms used in the SKAE extension specification are:

- A Subject Key is an asymmetric key pair public portion which is used in certificates or other verifiable cryptographic structures.
- An Attested Subject Key is TPM originated and resident non-migratable or CMK subject key certified by AIK.
- A Certified Credential is a public-key certificate issued by a Classic-CA to an end entity, where the public-key included into the certificate has been cryptographically bound to an AIK and it includes enough information for the relying party to validate that binding.

Although currently worked on as a separate specification, this document may be merged into the Credential Profile specification.

9 IWG Building Blocks

The IWG has adopted the Building Blocks (BB) approach as a way to identify, define and develop structures that are common across various use case scenarios. Some of these building blocks may employ features that are inherent and internal within a trusted platform (e.g. Integrity Measurement). Other may require communications with an entity external to the platform, and as such may involve the use of other building blocks and other protocols.

In this section each of the building blocks are described. In each section, besides the short *description* of the BB, *aspects and issues* that are relevant in the context of the IWG Architecture are provided, together with some *references* whenever necessary.

9.1 Integrity Measurement (BB1)

9.1.1 Description

TCG Glossary: *Integrity Measurement (Metrics): The process of obtaining metrics of platform characteristics that affect the integrity (trustworthiness) of a platform; storing those metrics; and putting digests of those metrics in shielded locations (called Platform Configuration Registers: PCRs).*

9.1.2 Aspects/Issues

Integrity Measurement is a building block core to the notion of trusted platforms and is used for platform authentication (BB10), which in-turn is the basis for a number of other building blocks composing multiple use cases.

Depending of the use case, Integrity Measurement may span from components at the hardware layer to components (e.g. software) at the applications-layer. Measurement at each layer is based on transitive trust rooted at the Root of Trust at the TPM.

9.1.3 References

See Measurement Logs specifications.

9.2 Integrity Storage (BB2)

9.2.1 Description

TCG Glossary: *Integrity Storage: Storage of integrity metrics in a log and storage of a digest of those metrics in PCRs.*

9.2.2 Aspects/Issues

The results of the process of Integrity Measurement must be stored in the Stored Measurement Log (SML) and the corresponding digests (of the metrics) in the PCRs.

9.2.3 References

See Measurement Logs specifications.

9.3 Integrity Reporting (BB3)

9.3.1 Description

TCG Glossary: *Integrity Reporting: The process of attesting to the contents of integrity storage.*

9.3.2 Aspects/Issues

For platform authentication of the Requestor to the Verifier, integrity reporting results needs to be conveyed across the two platforms, which may or may not reside in separate domains. The integrity reporting result must be signed using the AIK of the Requestor.

Two kinds of attestations need to be communicated from the Requestor to the Verifier:

- Identity of the Requestor using AIK-certificate or DAA-Credential (DAA-Signature).
- Attestations regarding the trusted platform of the Requestor

In addition to the AIK-certificate, in order for the Verifier to identify which Platform-CA issued the AIK-certificate a copy of the Platform-CA certificate (or pointers to it) may be communicated by the Requestor to the Verifier.

The definition of the Identity information and the Attestations used in cross-platform integrity reporting should be done independent of the authentication protocol (or other communications protocol) executed between the Requestor and Verifier platforms.

9.3.3 References

See TLS-Attestations specifications.

9.4 Evaluation of Integrity metrics (BB4)

9.4.1 Description

Evaluation of Integrity Metrics refers to the parsing and semantic evaluation of the integrity measurements of a Requestor by a Verifier, driven by the policies negotiated by both sides.

9.4.2 Aspects/Issues

The evaluation (by the Verifier) of the integrity metrics (from the Requestor) depends at the highest level of granularity on which specific PCR values the Verifier wishes to evaluate and what other information (e.g. AIK-credentials) requested by the Verifier. That is, the evaluation criterion is dependent on the Policies set by the Verifier. At least two approaches can be adopted as to how these information elements can be communicated:

- In-band (negotiated): The Verifier communicates (negotiates) the list of PCRs and Attestations it wishes to obtain from the Requestor through (during) the authentication protocol exchange (e.g. within TLS-Attest flows).
- Out-of-band: The choice of PCRs and Attestations is established by the Verifier and made known to the Requestor *prior* to the Requestor performing platform authentication to the Verifier. This information must be available to be accessed by the Requestor prior (and during) the process of the platform authentication. For example, as part of the TLS-Attest handshake, the Verifier may send a pointer (e.g. URL) to the Requestor where a (static) preferences file is stored. Alternatively, for intra-Enterprise scenarios the choice may be either coded within a configuration file set by the IT administrator (or the file could be resident on the local network).

The choice of PCRs and Attestations as well the choice of the means to communicate these selections, are set by policy (see BB7).

The evaluation of integrity metrics is considered as a building block because it is a basic operation common to all platforms (e.g. PC-client, server, Mobile) and fundamental to platform authentication.

9.4.3 References

See the TLS-Attestations specifications.

9.5 Response Actions (BB5)

9.5.1 Description

A Response-Action is specific to a given use case deploying trusted platforms. A response-action is tied to a given transaction between a Requestor and a Relying Party, and should be the result of an action or assertion originating from the Requestor at some earlier time.

9.5.2 Aspect/Issues

In general, there are some features of a Response-Action which makes it a BB:

- *Identifiable*: A Response-Action by a Relying Party must be identifiably connected to a given Action by a Requestor in an identifiable transaction between the Requestor and the Relying Party, possibly mediated or assisted by the Verifier. A Response-Action cannot emerge on its own, but must be the result of some earlier Action by the Requestor.
- *Atomic*: A Response-Action must be a unit of response and/or action that has an identifiable beginning and end. That is, for a pair of Requestor and Relying Party it must be clear that one Response-Action for a given transaction is different from another Response-Action for a second transaction, even though the two transactions and Response-Actions may be out of order.
- *Event/Assertion*: A Response-Action can be an *event* or *assertion*, specific to the use-case. Thus, for example, in the context of 802.1X a response-action can be the opening of a port at the Relying Party (i.e. an event), or it can be an assertion sent to the Requestor that authentication process failed.
- *Intra- or Inter-domain*: A Response-Action can be an event or assertion whose effects are intra-domain (R1 in Figure 9) to the domain Relying Party, or it can be an event or assertion which is inter-domain (R2 in Figure 9) to the domain of the Requestor.

9.6 Enforcement of Response-Actions (BB6)

9.6.1 Description

The enforcement of a Response-Action is specific to a given use case deploying trusted platforms.

9.6.2 Aspects/Issues

Since the Response-Action is emanating from a Relying Party (as a reaction or result of a request or assertion previously coming from a Requestor) within the context of a transaction, the enforcement of the Response-Action must refer to and be meaningful in that transaction:

- *Identifiable*: The Enforcement of Response-Action must refer to the transaction as a whole that initiated the Response-Action. Thus, when a Requestor is dealing with multiple

transactions, it must be able to identify the transaction to which a Response-Action applies, and enforce it.

- *Compatible expressions*: Since both a Requestor and a Relying Party can be enforcement points (R1 or R2 in Figure 9), both must employ the same (or compatible) policy languages or expressions to enforce a desired Response-Action. For example, in 802.1X the Authenticator (AP) must enforce the Response-Action emanating from the Authentication Server (AS), in a transaction initiated by the Supplicant. As such, all three must deploy the same policy architecture and language to express the desired effect.
- *Completeness/Error reporting*: Depending on the complexity of the Response-Action, its enforcement may or may not be complete (i.e. successful). In transactions whose Response-Action have a direct and visible effect on the Requestor (e.g. port open to a Supplicant in 802.1X), the success and completeness of the enforcement act is implicit through the successful effect. However, in a different type of transaction the enforcement point must explicitly report error cases and exceptions, particularly if the completeness of the entire transaction depends on the Response-Action being successfully enforced. For example, in an Airline Reservations system based on Web Services, failure to the deduct mileage-points for a frequent-flyers based ticket reservation should be reported explicitly since the entire transaction relies on this (successful) outcome.

9.7 Policy and Policy Authoring for Verifiers (BB7)

9.7.1 Description

In the context of platform authentication that makes use of TPM features, one key requirement is that a Verifier is able to understand and evaluate the integrity measurements of a client. As such, for each platform (e.g. PC, server, PDA) some policy language and policy authoring method must be used to express the integrity metrics that may be of interest to a Verifier.

9.7.2 Aspects/Issues

There are levels of policies that need to be distinguished within specific uses cases. Thus, it is useful to make a distinction between:

- *Platform-specific Integrity Policies*: For a given Trusted Platform (e.g. PC, server, PDA), there will be a set of integrity metrics that will be relevant to that platform. These will be independent of the Use-Case applications that make use of the metrics, though may be input to higher level policies governing the application.
- *Uses-Case Policies*: These are Use-Case specific policies (e.g. web-services, 802.1X) that govern a given Use-Case application (e.g. web forms, Radius access control) and are enhanced considerably by the availability of the underlying platform-specific integrity measurements. Thus, for example, using platform-specific integrity measurements, the Radius access control policy in 802.1X can now specify that certain platforms with a given set of PCR values must be given a specific set of IP addresses and be placed into a different VLAN.

9.8 User Authentication (BB8)

9.8.1 Description

User authentication is a building block that is common for all uses cases deploying trusted platforms. When a user (either an end-user or the Owner of a platform) seeks to gain access to applications whose security is based on capabilities of trusted platforms, then user authentication in itself becomes an important component of the whole value proposition of trusted platforms.

9.8.2 Aspects/Issues

Since a trusted platform can be viewed as an entity within its own right in a trust ecosystem, user authentication can be seen as consisting of two aspects:

- *Authentication of the user to a trusted platform*: Here, a user that has been authorized by the Owner of the platform is mechanically authenticated by the designated platform. The mechanics or protocol used by the platform must be set by the Owner of the platform. As such, it is the Owner that pre-select the type of credential the user must posses to gain access to the platform.
- *Authentication of a user (on a TP) by separate trusted platform*: Transitive trust may or may not be applicable to the case of user authentication to a remote platform. That is, after a user has been authenticated by his/her TP, the evaluation-result by that TP can be communicated to a remote TP which accepts that evaluation. In this way, the user need not be authenticated each time it seeks to access a remote TP.

Note that in general, the TPM could be used to store user credentials which will be accessed/used by the user-authentication system on the TP.

9.8.3 References

The User Authentication Working Group in the TCG is addressing User Authentication aspects in the context of Trusted Computing.

9.9 User Authorization (BB9)

9.9.1 Description

The term *authorization* is a technical term in the context of TPM activation and platform ownership. More broadly and abstractly, the term refers to the granting of privilege in relation to access to some resources. This implies, therefore, that some authority (possibly self authority) exists to grant authorization.

9.9.2 Aspects/Issues

Whereas authentication pertains to identification, authorization pertains to rights associated to that identity. The process of authorization should be distinct from that of authentication. Authentication as a process should provide some assurance with regards to the true identity of an entity (e.g. person). Authorization as a process provides assertions regarding resources available for access by the entity. The enforcement of authorization assertions upon an entity is often loosely referred to as *access control*. If entity has multiple identities (*roles*), then often authorization is in relation to the role taken-up by that entity for a specified time.

9.10 Platform Authentication (BB10)

9.10.1 Description

Platform Authentication is one of the most important building blocks as it provides a way for one platform to ascertain the integrity status of another platform based on TCG technology. The basic model for platform authentication is one consisting of the Requestor, Relying Party and the Verifier. The Requestor is seeking some action or outcome from the Relying Party, who must rely on the evaluation by the Verifier of the Requestor's platform.

9.10.2 References

See Section 4.

9.11 Sealing Keys to Configurations (BB12)

9.11.1 Description

Within a Trusted Platform that contains a TPM, a piece of data can be sealed (i.e. encrypted and signed) a manner that is only decryptable on the same system. In addition, the seal function in the TPM allows software to explicitly state the future "trusted" configuration that the platform must be in for the encrypted data to be revealed. This future configuration implicitly includes the relevant PCR values on the platform when the sealing operation was performed. Which PCR registers are going to be part of the seal operation is specified by the PCR composite object selected before the sealing operation.

When an unsealing operation succeeds, typically some proof is returned (to the caller of the operation) regarding the platform configuration during which the earlier sealing operation was performed. This proof is useful for some Use Cases.

For example, in the case of VPN-access based on a shared symmetric key (e.g. for IPsec), the VPN-Server may request the VPN-Client to seal the shared key to a given configuration tat he client. This means that next time the VPN-Client seeks to authenticate itself to the VPN-Server, the Server may ask the Client to unseal the encrypted shared key.

Other examples include sealing a key to decrypt a remote database or file.

9.11.2 References

See TSS Data_Seal and Data_Unseal commands in the TSS Specification document.

9.12 Platform Identity Registration (BB13)

9.12.1 Description

Platform Identity Registration is the act by a Trusted Platform of proving itself to be a trusted platform to a Trusted Third Party (TTP), and obtaining an Identity unique to that platform. If the TTP is a Platform-CA, then the identity takes the form of an AIK-Credential. Note that for some environments, it is not necessary that a Platform-CA be in existence. For example, within an Enterprise use case, an IT Administrator could maintain a database of Trusted Platforms within its domain, associating a unique identity to each platform. Another approach is to use the DAA protocols instead of a Privacy-CA, engaging the platform with a DAA-Issuer and DAA-Verifier.

9.12.2 References

See Section 4.7.

9.13 Key Migration/Backup (BB14)

9.13.1 Description

Key Backup and Migration is a core building block to the operational infrastructure supporting trusted platform, as the function is used by many other building blocks and protocols. Backup refers to the safe storage of the (sealed) platform keys to a different location, with the intent of the keys being accessible (unsealable) to the same platform. Migration refers to the moving of the platform keys from an old platform to anew platform, possibly sealing keys and user data to the new platform.

9.13.2 References

For a brief discussion see Section 8.4. For details, the reader is directed to the Backup/Migration specifications.

9.14 Secure Time Stamping (BB15)

9.14.1 Description

Secure Time Stamping is an important building block because many functions and services within a trusted ecosystem relies on the availability of a trustworthy source of time. Examples of uses case of trusted platforms with secure time stamping requirements include credentials management (e.g. certificate expiration checks), content management (e.g. content dead at a given time) and Internet monetary transactions (e.g. auctions). There is an opportunity for the use TCG technology itself (e.g. time-servers) to secure time-stamping protocols, thereby providing higher assurance of the reliability of the time-stamps.

9.14.2 References

See RFC1305, RFC2030 and STIME Working Group in the IETF.

9.15 Platform Identity Credential Revocation (BB16)

9.15.1 Description

There are some circumstances in which an AIK-Credential needs to be revoked, prior to its expiration time. Examples include the mishandling of the AIK-private-key (e.g. during its issuance from the Platform-CA), incomplete erasure of a TPM, clone detected by either the Platform-CA or Verifier, and other unintended releases of the AIK-private-key by the Owner. Although the simple erasure of the AIK-private-key may be sufficient in many cases to render the AIK-Credential unusable, it is best practice today for Certificate Authorities to perform a complete retirement of certificates (i.e. revocation) when it is suspected that some security-related problems have occurred. Similar requirements also exist in the case of DAA and DAA-Credentials.

In the context of Trusted Platforms, a finer grain of conditions for revocation of AIK-Credentials may be required. For example, when a closed (private) Platform-CA is the issuer of an AIK-Credential and the AIK-Credential is inadvertently release to the public, revocation may not be necessary. The semantics for revocation should be tied to what the credential attests, and the policy of usage of the credential. As such, a scoring approach may be used to rank the importance of the credential in a given use case of the Trusted Platform.

9.15.2 References

IETF PKIX RFCs.

9.16 Hardware-rooted Application key lifecycle (BB17)

9.16.1 Description

The notion of public-keys that are application-specific and which are rooted in hardware provides attractive possibilities for providing a higher security to those applications. The TPM_CertifyKey command could be used to assert this fact. Examples of applications include digital-signatures with keys/certificates rooted in the signer's hardware, and server keys/certificates that are bound to the server hardware. Similar approaches can be found in HSM units today.

Since the applications keys and certificates are rooted in TPM hardware, a complete lifecycle for managing the application keys is required, which may resemble the lifecycle of the keys and certificates in Trusted Platforms.

Since there are a number of ways to achieve the above effect for each use case, it is recommended that limitations and procedures be defined in the TCG Best Practices document.

9.16.2 References

See TPM_CertifyKey command and related structures. See TCG Best Practices document.

9.17 Atomicity (BB18)

9.17.1 Description

Atomicity in Trusted Computing refers to the *instance-uniqueness* of (migratable) keys and credentials related to a given platform. That is, some keys are defined to exist only on one platform at any one time.

The need for atomicity is particularly relevant in the context of the Backup/Migration in which a given migratable key could potentially exist in three places: (i) An “old” platform, (ii) a BackUp/Migration Server, and (iii) a “new” platform. As such, in performing function and services that may cause replication of certain migratable keys, atomicity must always be observed as a matter of rule and according to the policies governing the given use-case.

9.17.2 References

See the IWG Backup/Migration specification.

9.18 Provenance (BB20)

9.18.1 Description

Provenance of keys and credentials in Trusted Computing refers to the history of existence of the keys and credentials. Provenance is of particular interest to “consumers” of keys and credentials in the Trusted Platform Lifecycle. One such “consumer” is the Platform-CA.

When a Platform-CA wishes to issue an AIK-Credential associated to a given platform, it must evaluate the EK-Credential bound to the EK-Private-key present in the platform’s TPM and evaluate the Platform-Credential associated with the given platform. As such, the Platform-CA would take notice of the origins of the EK key pair (e.g. was the EK-private-key generated in the TPM, or was it injected), the issuer of the EK-Credential and the CPS of that issuer, and other aspects of the EK-Credential and Platform-Credential.

Besides the Credentials themselves, other supporting information (e.g. signed manifest) could represent the repository of provenance-related information pertaining to the keys and credentials of the platform.

9.18.2 References

See IWG Credentials Profile specification.

9.19 EK/Platform Credential Issuance (BB21)

See Section 3.3 and 6,and see the IWG Credentials Profile specification.

9.20 Platform deployment and initial setup (BB22)

See Section 3 of the current document.

10 References

[1] Trusted Computing Group, *Infrastructure Use Cases*, Specification Version 1.0, Revision 0.25, February 2004, Work in Progress.

[2] Trusted Computing Group, *Credential Profile for v1.1b*, Specification Version 1.0, June 2005, Work in Progress.

[3] Trusted Computing Group, Backup and *Migration Services Specification*, Specification Version 1.0, Revision, June 2005, TCG Published.

[4] Trusted Computing Group, *Core Integrity Schema*, Specification Version 1.0, Revision 0.7, June 2005, Work in Progress.

[5] Trusted Computing Group, *Subject Key Attestation Evidence (SKAE) Extension*, Specification Version 1.0, June 2005, TCG Published.

[6] Trusted Computing Group, *TLS Extensions for Attestation*, Specification Version 1.0, Revision 0.8, July 2004, Work in Progress.

[7] Trusted Computing Group, *Credential Profile for v1.2*, Specification Version 1.2, Revision 0.1, June 2004, Work in Progress.

[8] Trusted Computing Group, *DAA Summary*, September 2003, Work in Progress.

[9] Trusted Computing Group, *TLS Extensions for DAA Attestation*, Specification Version 1.2, Revision 0.1, July 2004, Work in Progress.

[10] Trusted Computing Group, *TPM Specifications v1.2*, October 2003.

[11] Trusted Computing Group, *TSS Specifications v1.1*, August 2003.

[12] Trusted Computing Group, *TCG Design Guidelines, (Glossary section)*, June 2004. Work in Progress.

[13] M. Myers, X. Liu, J. Schaad, J. Weinstein, *Certificate Management Messages over CMS*, RFC2797, Standards Track, April 2000, IETF.

[14] P. Hallam-Baker (ed.), *XML Key Management Specification (XKMS 2.0)*, Version 2.0, Candidate Recommendation 5 April 2004, W3C. (http://www.w3.org/TR/2004/CR-xkms2-20040405)

[15] M. Myers, C. Adams, D. Solo, D. Kemp, *Internet X.509 Certificate Request Message Format (CMC)*, RFC2511, Standards Track, March 1999, IETF.

[16] R. Housley, *Cryptographic Message Syntax (CMS)*, RFC2630, Standards Track, June 1999, IETF.

[17] Housley, R., Ford, W., Polk, W. and D. Solo, *Internet X.509 Public Key Infrastructure: Certificate and CRL Profile*, RFC2459, Standards Track, January 1999, IETF.

[18] S. Chokhani, W. Ford, *Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework*, RFC2527, Informational, March 1999, IETF.

[19] R. Housley, W. Polk, W. Ford, D. Solo, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, RFC3280, Standards Track, April 2002, IETF.

[20] S. Farrell, R. Housley, *An Internet Attribute Certificate Profile for Authorization*, RFC3281, Standards Track, April 2002, IETF.

[21] D. Pinkas, J. Ross, N. Pope, *Electronic Signature Formats for long term electronic signatures*, RFC3126, Informational, September 2001, IETF.

[22] E. Brickell, J. Camenisch, L. Chen, *Direct Anonymous Attestations*, In Proceedings of 11th ACM Conference on Computer and Communications Security, ACM Press, 2004.