Digital Trust - Trusted Computing and Beyond A Position Paper*

Raja Naeem Akram, and Ryan K. L. Ko
Cyber Security Lab, Department of Computer Science, University of Waikato
Hamilton, New Zealand.
{rnakram, ryan}@waikato.ac.nz

***Abstract*—Along with the invention of computers and interconnected networks, physical societal notions like security, trust, and privacy entered the digital environment. The concept of digital environments begins with the *trust* (established in the real world) in the organisation/individual that manages the digital resources. This concept evolved to deal with the rapid growth of the Internet, where it became impractical for entities to have prior offline (real world) trust. The evolution of *digital trust* took diverse approaches and now trust is defined and understood differently across heterogeneous domains. This paper looks at digital trust from the point of view of security and examines how valid trust approaches from other domains are now making their way into secure computing. The paper also revisits and analyses the Trusted Platform Module (TPM) along with associated technologies and their relevance in the changing landscape. We especially focus on the domains of cloud computing, mobile computing and cyber-physical systems. In addition, the paper also explores our proposals that are competing with and extending the traditional functionality of TPM specifications.**

I. Introduction

Reliance on computers and related services has increased since their inception. Computers now form large interconnected networks that consist of sub-networks managed and operated by divergent organisations. In a restricted network environment, where all participants were vetted (offline) beforehand, managing security and trust was comparatively straightforward. However, with increased complexities in networked systems and participants, offline vetting became impractical in some cases. Therefore, technological means were required to assess and associate trust with individual entities. In this paper, the trust established using technological means, in whole or in part, is referred to as **"digital trust"**.

Similar to the diversification of computer systems, digital trust also came in several different incarnations. Each computing domain defined and articulated the notion of digital trust in a specific manner that satisfied its requirements. Therefore, the assumption that digital trust will have a single definition is difficult to substantiate. Individual definitions of digital trust might be valid in their respective domains. However, issues arise when a definition of trust in one domain is applied to a different domain without adequately adjusting it, as for example, in the notion of digital trust related to computer security via provenance [1, 2]. Such concerns are rare but nonetheless exist.

*This work is a position paper that discusses notion of trust in different computing fields. It also analyses our proposals for emerging fields.

In the field of information security, the measurement and validation of digital trust and trustworthiness play crucial roles. The foundation of secure and trusted computing[1] can be argued to be based on the effectiveness of digital trust evaluation and validation mechanisms [7]. With increased reliance on on-demand and ubiquitous services, connecting with a wide range of devices that might not be under the control of a particular organisation / individual is becoming commonplace. In such a chaotic and ever-changing environment, dynamic establishment and verification / validation of digital trust is crucial. The need for digital trust will only grow with increasing adoption of cloud computing, mobile platforms and Cyber-Physical System (CPS).

A. Contribution of Paper

In this paper, to give an overview of digital trust, we revisit the variance in the notion of digital trust in selected domains of computer science. The crux of the paper is the argument that digital trust in security and privacy services provides secure and trusted computing — which is difficult to achieve without a trusted entity/platform. Furthermore, system architects and implementors should also understand the subtle differences between the notion of the trust in security and other computing domains. This paper also provides a brief survey of the trust and trustworthiness in the context of security and privacy services. In addition, the paper evaluates the state-of-the-art of trusted computing, issues related to its slow adoption and whether traditional trusted computing is relevant to future computing paradigms. It also presents the challenges faced by digital trust for security and privacy services in mobile environments, cloud computing [8] and embedded platforms (Internet of things [9], Cyber-Physical Systems [10]). We provide a brief description of our proposals for potential future incarnations of digital trust in the field of trusted computing, along with open research questions.

B. Structure of Paper

Section II explores the definition of digital trust across a selected subset of computer domains. In section III, the paper discusses the trusted computing initiative, associated

[1]Secure and Trusted Computing: This term refers to the efforts made toward enabling technologies to ascertain trust in a device's state and security in a distributed environment. For example, Trusted Computing Group (TCG) [3, 4], ARM TrustZone [5] and M-Shield [6].

expectations and the reality of its slow adoption. Subsequently, in section IV the paper evaluates the relevance of trusted computing in future computing paradigms. In section V, our proposed potential future architectures for trusted computing are discussed. Finally in section VI, we list open research questions and conclude the paper.

II. Digital Trust

The definition of trust, taken from Merriam Webster's online dictionary[2] states that trust is a "belief that someone or something is reliable, good, honest, effective, etc."

Based on this, we generically define digital trust as "a trust based either on past experience or evidence that an entity has behaved and/or will behave in accordance with the self-stated behaviour." The self-stated purpose of intent is provided by the entity and this may have been verified/attested by a third party. The claim that the entity satisfies the self-stated behaviour can either be gained through past interactions (experience) or based on some (hard) evidence like validatable / verifiable properties certified by a reputable third party (i.e. Common Criteria evaluation for secure hardware [11]). This definition is not claimed to be a comprehensive definition for digital trust that encompasses all of its facets. However, this generic definition will be used as a point of discussion for the rest of the paper.

In subsequent sections, we discuss the notion of digital trust in different (selected) computer domains to show the variations in the definition and application of trust.

A. Semantic Web

The semantic Web is the framework and associated architecture that enables streamlined and flexible data sharing, so the data can be machine processable, and consumed by heterogeneous applications, beyond the boundaries of the enterprises and communities from where the data originated [12]. The notion of trust is a core component of the semantic Web [13, 14] and in his earlier publications, Tim Berners-Lee [14, 15] included the trust layer in the sematic Web stack (see Figure 1) since its inception. Evidently, trust is not an afterthought in the domain of the semantic Web.

To accept and rely on a source of data in the semantic Web, consumers (applications and/or users) have to establish a notion of trust in the data and possibly also in its source. The information presented as part of the semantic Web should be treated as "claims," not facts [16]. Digital trust in the semantic Web provides a level of confidence in a piece of information. Proposed architectures for digital trust in the semantic Web include reputation-, context- and content-based mechanisms.

The reputation-based mechanism uses rating mechanisms, where either the information consumers or other websites rank their confidence in an information source [13, 17]. These include, for example, the rating systems used by eBay and IMDB. A reputation-based mechanism requires information consumers to consistently provide ratings, which might become an unnecessarily tedious and potentially bias-filled task for them.

[2]Website: http://www.merriam-webster.com/dictionary/trust

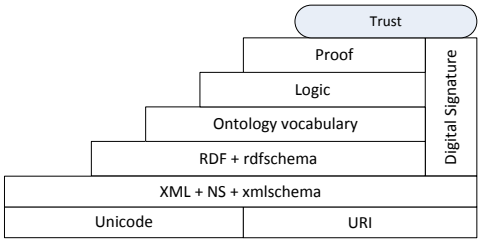

Figure 1. Semantic Web Stack [15]

In context-based mechanisms, trust is based on the meta-information that details the circumstances in which the information was provided [18]. The context stipulates meta-information like who, what, when, where, why and how. It also takes into account the authorship of the information. For example, a product description published on its manufacturer's website will be trusted more than one published on a competitor's website. Another example is only accepting election results from an official election organising body rather than from any news sources.

Content-based trust mechanisms rely on the content of information and trust rules/axioms [19] and may require information to be presented by a number of notable and independent resources (e.g. news outlets). The basic construct of trust in the semantic Web is to establish the validity of the presentation of openly accessible information.

B. Data Provenance

Data provenance can be understood as a snapshot of all the transformations a data item has gone through during the process that created the data item. In other words, as defined by [20, 21] provenance is the meta-data of the derivation history of data. Data provenance is an important component in many data-intensive studies and/or industries like eScience [22] and healthcare [23]. In such environments, provenance ensures the quality of data and repeatability of results.

In the area of data provenance, trust is measured and associated based on the history of the data. Trust measurement is affected by questions like "who created the data," "who processed/modified it," "what is the current state of the data" and "(prior) trust in the data creator and processing entities" [24, 25]. We can generalise that trust and measurement of trustworthiness in a combined approach that includes reputation (reputation of data sources and handlers), context (operations performed on the data) and content (current state of the data).

In some proposals [24] the security of the data originators/handlers is included, although the security evaluation is not described. However, in a number of works related to trust in data provenance, data quality and security of provenance records are the main focus, not data security [26, 27].

C. Secure and Trusted Computing

In the real world, trust in an entity is based on a feature, property or association that is entailed in it. In the computing world, establishing trust in a distributed environment also follows the same assumptions. The concept of trusted platforms is based on the existence of a trusted and reliable device that provides evidence of the state of a given system. How this evidence is interpreted is dependent on the requesting entity. Trust in this context can be defined as an expectation that the state of a system is as it is considered to be: secure. This definition requires a trusted and reliable entity called a Trusted Platform Module (TPM) to provide trustworthy evidence regarding the state of a system. Therefore, a TPM is a reporting agent (witness) not an evaluator or enforcer of the security policies. It provides a root of trust on which an inquisitor relies for the validation of the current state of a system.

The TPM specifications are maintained and developed by an international standards group called the Trusted Computing Group (TCG)[3] Today, TCG not only publishes the TPM specifications but also the Mobile Trusted Module (MTM), Trusted Multi-tenant Infrastructure, and Trusted Network Connect (TNC). With emerging technologies, service architectures, and computing platforms, TCG is adapting to the challenges presented by them.

1) Trusted Platform Framework: The basic framework for the trusted platform is to have a root of trust (preferably in hardware) and trust in it is necessary if an entity has to measure the trustworthiness of a system. The root of trust in the TCG specifications [3, 4] combines Root of Trust for Measurement (RTM), Root of Trust for Storage (RTS), and Root of Trust for Reporting (RTR). The RTM is an independent computing platform that has a minimum set of instructions, which are considered to be trusted for measuring the integrity matrix of a system. On a typical desktop computer, the RTM will be part of the BIOS (Basic Input Output System) and in this scenario, it is referred to as the Core Root of Trust for Measurement (CRTM). Where the RTS and RTR are based on an independent, self-sufficient, and reliable computing device that has a pre-defined set of instructions to provide authentication and attestation functionality, such a device is referred to as a Trusted Platform Module (TPM).

A platform can be considered a trusted platform if it has a TPM and supporting architecture for the "Trusted Building Block" (TBB). The TBB includes CRTM, physical connection between a CRTM and the motherboard (of the platform), connection between a TPM and the motherboard, and functionality to detect physical presence. Physical presence implies the direct interaction of a user with the platform, which is traditionally based on a secret credential that in theory is only known to the user. By verifying the credentials, the platform assumes that the platform owner is physically present. Figure 2 illustrates the trusted platform framework.

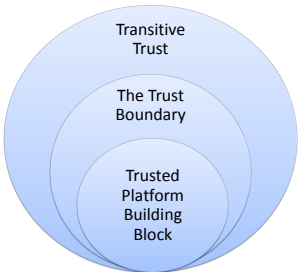

Figure 2. Trusted Platform Framework [28]

The trust boundary is a collection of the TBB and roots of trust. A TPM extends the trust from roots of trust through transitive or inductive trust. A transitive trust is a process that enables a root of trust to provide a trustworthy description (e.g. hash generation) of a second function (e.g. software). The requesting entity can then verify whether it can trust the second function based on the description provided by the relevant TPM. The rationale behind transitive trust is that if an entity trusts the TPM of a platform, it will also trust its measurements.

In this section, the discussion of secure and trusted computing mainly focused on the TPM. There are other proposals for secure and trusted computing but none has the status of the TPM specifications. We will discuss a few of these proposals in later sections, to contrast with the TPM architecture.

D. Trust and Trustworthiness

From the discussion in this section, we can delineate two distinct types of trust frameworks: *hard trust* and *soft trust*. The term *hard trust* refers to architectures that base the measurement/foundation of trust on verifiable and independently validated hardware (e.g. TPM [4], ARM TrustZone [5]). In contrast, the term *soft trust* is associated with trust measurement and assessment mechanisms that do not rely on trusted hardware: examples of soft trust can be reputation-, context- and content-based trust mechanisms.

Hybrid trust combines soft and hard trusts to provide a potentially comprehensive approach. In the field of security, a substantial number of trust proposals can be categorised as hard trust. This is not to say that soft trust might not be valid or applicable to the security domain [29, 30]. However, soft trust on its own might not be a preferable approach to progressing with secure and trusted computing. In the rest of the paper, we discuss hard-trust based mechanisms for secure and trusted computing.

Whether a *soft*, *hard* or *hybrid* trust approach is used, it can be divided into two parts. First is the trusted measurement and reporting framework and second is the mechanism to generate a score. The generated score will represent the trustworthiness of the relevant entity/information. Data provenance mechanisms can be used to measure and report the state/quality of the data [24, 31]. Based on these measurements and reports,

[3]Trusted Computing Group (TCG) is the culmination of industrial efforts that included the Trusted Computing Platform Association (TCPA), Microsoft's Palladium, later called Next Generation Computing Base (NGSCB), and Intel's LaGrande. All of them proposed how to ascertain trust in a device's state in a distributed environment. These efforts were combined in the TCG specification that resulted in the proposal of TPM.

trustworthiness can be calculated; however, data provenance does not become part of the calculations that ascertain the trustworthiness of data. Similarly, TPM is a trusted and secure measurement and reporting hardware system, where the calculation of the trustworthiness of a system is left to the inquirer (i.e. the entity that requests the integrity report from the TPM) [32].

Therefore, security and reliability of the trust measurement and reporting agent are as crucial as the trustworthiness of the system. The basic premise is the invariability and effectiveness of the measurement and reporting mechanism even when in the control of a malicious entity. If a malicious entity can influence the trust measurement and reporting mechanisms then calculation of trustworthiness is of no value. For this reason, hard trust is usually the preferred choice for providing proof that the trust measurement and reporting mechanism is reliable and tamper-resistant, satisfying the requirement for an effective mechanism even when controlled by an active adversary. The calculation of trustworthiness is dependent on the evaluator and it might be independent of the trust architecture — except for mechanisms that integrate hard trust with reputation-based systems [29]. For example, if a malicious user accepts an untrusted system as trusted, then he/she is taking the risk. In such systems a malicious user can still report that system 'A' is untrustworthy even when the trustworthiness of system 'A' is high.

III. Trust in Security and Privacy

In this section, we briefly discuss the TPM and the Mobile Trusted Module (MTM). Subsequently, we discuss the initial promise of the trusted computing initiative and why in reality it did not get the traction that was expected. Finally, we evaluate the potential future of trusted computing.

A. Trusted Platform Module

The basic TPM architecture and its different components are shown in Figure 3. For in-depth discussion of individual components and their functionality please refer to [32, 33].

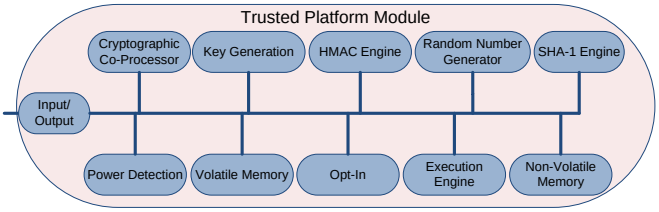

Figure 3. Generic Architecture of Trusted Platform Module [4]

To describe how the TPM measures and reports trust measurements, we restrict our discussion to measurement and reporting operations only.

1) Secure Boot (Measurement Operation): When a user boots up her computer, the first component to power up is the system BIOS (Basic Input/Output System). On a trusted platform, the boot sequence is initiated by the Core BIOS (i.e. CRTM) that first measures its own integrity. This measurement

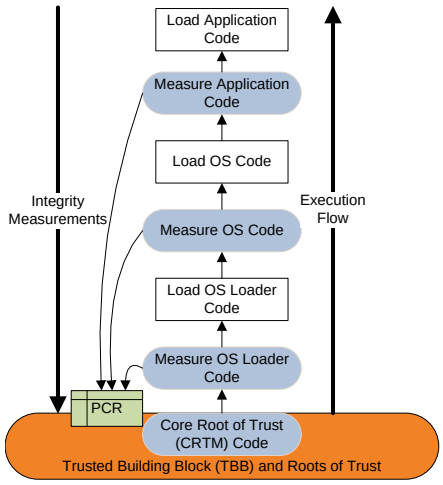

Figure 4. Trusted Platform Boot Sequence

is stored in PCR-0[4] and later it is extended to include the integrity measurement of the rest of the BIOS. The Core BIOS then measures the motherboard configuration setting, and this value is stored in PCR-1. After these measurements, the Core BIOS will load the rest of the code of the BIOS.

The BIOS will subsequently measure the integrity of the ROM firmware and ROM firmware configuration, storing them in PCR-2 and PCR-3 respectively. At this stage, the TBB is established and CRTM will proceed with integrity measurement and loading of the Operating System (OS).

The CRTM measures the integrity of the "OS Loader Code," also termed the Initial Program Loader (IPL) and stores the measurement in the PCR. The designated PCR index is left to the discretion of the OS. Subsequently, it will execute the "OS Loader Code" and on its successful execution, the TPM will measure the integrity of the "OS Code". After measurement is taken and stored, the "OS Code" executes. Finally, the relevant software that initiates its execution will first be subjected to an integrity measurement, and values will be stored in a PCR then sanctioned to execute. This process is shown in Figure 4, which illustrates the execution flow and integrity measurement storage.

By creating a daisy chain of integrity measurements, a TPM provides a trusted and reliable view of the current state of the system. Any software, whether part of an OS or an application, has an integrity measurement stored in a PCR at a particular index. If the value satisfies the requirement of the software or requesting entity, then it can ascertain the trustworthiness of the system or otherwise take action. As discussed before, a TPM does not make any decision: it only measures, stores, and reports integrity measurement in a secure and reliable

[4]Platform Configuration Register (PCR): A Platform Configuration Register (PCR) is a 160-bit (20 bytes) data element that stores the result of the integrity measurement, which is a generated hash of a given component (e.g. BIOS, operating system, or an application). Therefore, a group of PCRs form the integrity matrix. The process of extending PCR values is: $PCR_i = Hash(PCR'_i||X)$, where i is the PCR index, PCR'_i represents the old value stored at index i, and X is the sequence to be included in the PCR value. "||" indicates the concatenation of two data elements in the given order. The starting value of all PCRs is set to zero.

manner. When a TPM reports the integrity measurement, it is recommended that it should generate a signature on the value - avoiding replay and man-in-the-middle attacks [4].

2) Reporting and Attestation Operations: The attestation process, whether initiated by the relevant user/administrator/third-party either locally or remotely, involves the generation of a signature using the respective Attestation Identification Key (AIK) on the (associated/requested) PCR values [28]. The signature assures requesters of the validity of the integrity measurement stored in the PCRs. The choice of the AIK and PCR index is dependent on the respective user, platform (OS) or application.

B. Mobile Trusted Module

The growth of mobile computing platforms has encouraged TCG to propose the Mobile Trusted Module (MTM). In this section, we briefly discuss MTM architecture and operations, along with how the MTM differs from the TPM.

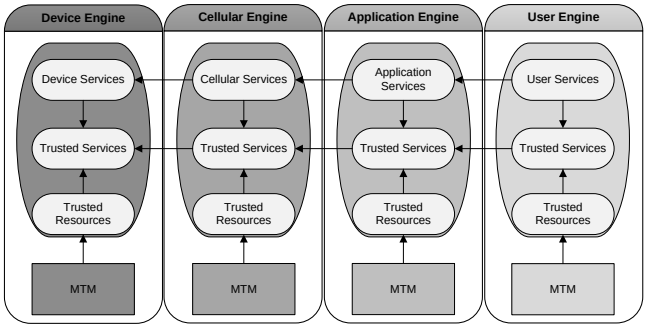

Figure 5. Possible (Generic) Architecture of Mobile Trusted Platform

1) Basic Architecture and Operations: The ecosystems of mobile computing platforms (e.g. mobile phones, tablets, PDAs) are fundamentally different from traditional platforms. Therefore, while the architecture of the MTM has some features from the TPM specification, it introduces new features to support its target environment. The main changes introduced in the MTM that make it different from the TPM specification are stated below:

1) The MTM is required not only to perform integrity measurement during the device bootup sequence, but also to enforce a security policy that aborts the system from initiating securely if it does not meet the trusted (approved) state transition.
2) The MTM does not have to be in hardware: it is considered as a functionality, which can be implemented by the device manufacturers as an add-on to their existing architectures.
3) The MTM specification supports parallel instances of MTM, associated with different stakeholders.

The MTM specification [34] is dynamic and scalable to support the existence of multiple MTMs interlocked with each other as shown in Figure 5. The MTM refers to them as engines, where each of these engines is under the control

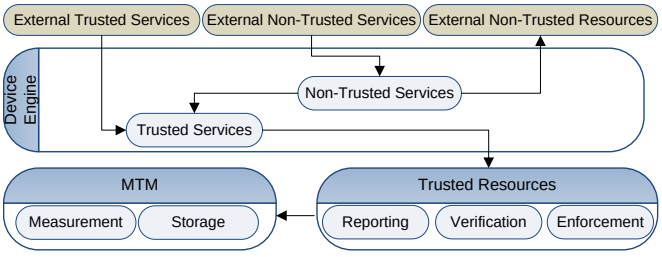

Figure 6. Generic Architecture of an Abstract Engine

of a stakeholder, including the device manufacturer (Device Engine), Mobile Network Operator (Cellular Engine), Application Provider (Application Engine), and User (User Engine); as illustrated in Figure 5. A point to note is that each engine is an abstraction of trusted services associated with a single stakeholder. Therefore, on a mobile platform there can be a single hardware system that supports the MTM functionality and is accessed by different engines.

Each abstract engine on a mobile platform supports: 1) provision to implement trusted and non-trusted services (normal services) associated with a stakeholder, 2) a self-test to ascertain the trustworthiness of its own state, 3) storage of TPM Endorsement Key (EK) (which is optional in MTM) and/or AIKs, and 4) key migration.

We can further dissect each abstract engine as a component of different services as shown in Figure 6. The non-trusted services in an engine cannot access trusted resources directly. They have to use the Application Programming Interfaces (APIs) implemented by trusted services. Trusted resources, including reporting, verification, and enforcement, are new concepts that are introduced in the MTM specifications. The MTM measurement and storage services shown in Figure 6 are similar to the TPMs discussed in previous sections.

The MTM specification defines two variants of the MTM profile depending upon who is the owner of a particular MTM. They are referred to as Mobile Remote Ownership Trusted Modules (MRTM) and Mobile Local Ownership Trusted Modules (MLTM). The MRTM supports remote ownership, which is held either by the device manufacturer or the mobile network operator, where MLTM supports the user ownership.

The roots of trust in the MTM include the ones discussed for TPM including RTS, RTM, and RTR. However, the MTM introduces two new roots of trust: Root of Trust for Verification (RTV) and Root of Trust for Enforcement (RTE). During the MTM operations on a trusted mobile platform, we can logically group different roots of trust; for example RTM and RTV are grouped together to perform an efficient measure-verify-extend operation as illustrated in Figure 7. Similarly, RTS and RTR can be grouped together to provide secure storage and trustworthiness of the mobile platform.

The MTM operations as shown in Figure 7 begin when a process starts execution, and they are listed below:

1) The RTM will perform an integrity measurement of the initiated process.
2) The RTM will register an event that includes the event data (application/process identifier) and associated in-

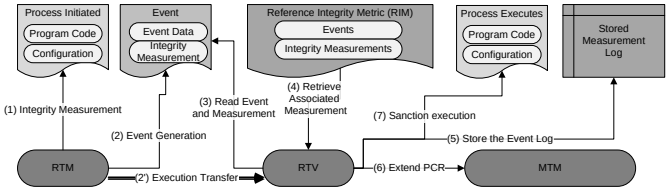

Figure 7. MTM Measurement and Verification Process

tegrity measurement. The RTM then transfers the execution to the RTV.

3) The RTV will read the event registered by the RTM.
4) The RTV will then search the event details from the Reference Integrity Metric (RIM), which includes the trusted integrity measurements associated with individual events, populated by the engine owner. This operation makes the MTM different from the TPM, as the later does not make any decision regarding the trustworthiness of the application or process. However, MTM does so via the comparison performed by the RTV to verify that the integrity measurement performed by the RTM matches the one stored in the RIM. If the integrity measurement does not match, the MTM will terminate the execution or disable the process. If the verification is successful then it will proceed with steps 5 and 6 along with sanctioning the execution (step 7).
5) The RTV will register the event in the measurement logs. These logs give the order in which the measurements were made to generate the final (present) value of the associated PCR.
6) The RTV will extend the associated PCR value that is stored in the MTM.
7) If verification is successful, it will sanction the execution of the process.

C. Trusted Computing: An Architecture on the Rebound?

Since its inception, the TPM has been the subject of adverse reaction from the community (i.e. common users, organisations and researchers). The issue with TPM initially revolved around the argument of "big" corporations trying to take the control of computers away from their owners. Such concerns were addressed by the TPM providing control of the TPM to the owner and significantly improving the overall privacy architecture for reporting mechanisms.

Another concern was based on the TPM design goals that stated it could provide a platform for potentially "secure and unhackable" Digital Rights Management (DRM) frameworks. This again raised concerns regarding the control that a computer owner exercises over her computer.

These two concerns overshadowed the potentially beneficial features of the TPM, like secure boot, secure storage and secure management of cryptographic keys. Increasing numbers of PCs now have built-in TPM, although a substantial number of these modules are not enabled by their respective users. Similarly, there are not many applications that use TPM features to provide added security and privacy services to their customers. Therefore, it seems that there are no strong reasons for an end-user to actively use the TPM for his/her security and privacy requirements.

With increasing reliance on computing platforms, the TPM will increasingly play a crucial role in providing secure computing platforms. In addition, general users are also becoming aware of the potential security issues that might lead them to activate their TPMs. However, while the adoption of the TPM may increase, the specification is not changing with the changing landscape of computing technologies. There are many competing technologies that provide similar services to TPM: examples include AEGIS [35], ARM TrustZone [5] and M-Shield [6]. Most of these technologies extend the traditional features of the TPM and provide runtime (execution) security and trust services.

In next section we will look at a few of the emerging spaces where TPM could play a role in providing secure and trusted computing environments. To accommodate the requirements of these emerging technologies, TPM specifications have to be flexible.

IV. Trusted Computing and the Changing Landscape

In this section, we discuss the emergence of mobile platforms, cloud computing and CPS along with their potential requirements.

A. Mobile Platforms

With the advent of smart phones and the concept of "Apps," the mobile platform is becoming the core device an end-user will use to access different services over the internet or in the real world. For example, mobile banking enables a user to access his/her bank account from a mobile phone. Similarly, Near Field Communication (NFC) [36] is enabling mobile phones to communicate directly with physical devices (e.g. access control points) without requiring the internet.

TCG proposed MTM as an important step, but its adoption is not widespread. In addition, the features and access required by many of the security-sensitive services on a smart phone, like EMV specifications (for debit and credit cards) [37] are not satisfied by the MTM. In sections V-A we discuss the our proposed architecture that can support trusted computing for mobile platforms.

B. Cloud Computing

Cloud computing is emerging as the 'next big thing' in the IT landscape. Cloud computing provides scalability, flexibility and elasticity to organisations that can provision resources depending upon the work load, thus efficiently managing their requirements. However, most cloud services are provided by third parties, which means that the data storage and processing will be off-site. In addition, most of the services provided to cloud users are virtualised. This might isolate the services from directly accessing the hardware, especially secure hardware (i.e. TPM).

For sensitive services like health care, moving to the cloud is not straightforward. They have to get the cloud service

provider to make an offline agreement (official contract governed by local regulations/laws of the country where the health care provider resides) regarding the security and storage of data [38].

Trusted computing for cloud services is mainly focused on the design of effective soft trust-based architectures [39, 40]. Other approaches advocate the collection of data provenance to increase the awareness of data in third-party cloud services [38, 41, 42]. However, the importance of trusted computing for cloud services is accepted. In addition, there is potential for trusted computing to become crucial to cloud computing as its adoption increases. In section V-B, we look at how hard trust-based mechanisms can be built for cloud computing environments.

C. Cyber-Physical Systems

Cyber-Physical Systems (CPSs) are collections of embedded devices that control/manage physical entities. CPSs are deployed in diverse fields including the automotive industry, civil infrastructure, energy, healthcare and consumer appliances. CPSs are being developed so that that they construct a collaborative architecture, not only among the embedded devices in a single CPS but also extending to inter-CPS collaborative architectures. Potentially, CPSs may out-number any other computing environment by a large margin in the near future [43].

The CPS will be ubiquitous in modern life and it has the potential to interconnect different computing environments with the physical world. Such an environment has the potential to be the prime target of malicious entities. Therefore, in order to provide reliable and efficiently protected services, trusted computing becomes a crucial part of the CPS [44]. In section V-C. We discuss the idea of trusted computing platforms for individual embedded devices, and of building a trusted environment from individual devices in the CPS to create a collaborative architecture of trust.

V. Beyond Traditional Trusted Computing Platforms – Our Proposals

In this section, we discuss our proposed architectures that go beyond the traditional notion of trusted computing and their applications in different computing domains.

A. Consumer Centric Trusted Device

The adoption of mobile phones and tablet-based computing platforms (e.g. iPads) is increasing. To some extent, the security and privacy issues of personal computers, including insecure execution environments, also apply to hand-held devices. As reliance on these devices increases, so will threats to the security and privacy of the platform and its users. For example, a healthcare mobile application, if it is badly designed and is subsequently compromised, may reveal users' sensitive medical information.

A possible solution is to have a tamper-resistant execution environment that executes a programme in a trusted, secure, reliable, and fault-tolerant environment. Among widely deployed tamper-resistant devices, two are prominent: the Trusted Platform Module (TPM) [33], and smart cards [45].

The TPM provides a platform's integrity measurement with cryptographic protection, in contrast to smart cards that provide a generic execution environment, in which an application can execute and store application code and data. This landscape maps from smart cards, mobile phones, tablets and general-purpose computers through to Machine-to-Machine communication and the Internet of Things [9]. It would be beneficial to have an interoperable unified architecture that provides a secure and reliable execution and storage environment for different computing devices.

A User-Centric Tamper-Resistant Device (UCTD) is a "one" end user device that provides a unified architecture [46]. Figure 8 shows different possible form factors for the UCTD, various applications that it can host, and different industries that can use the provided functionality. As UCTD is based on the User Centric Smart Card Ownership Model (UCOM) architecture [50], it requires the Trusted Execution & Environment Manager (TEM) [32] to provide security and trust assurance.

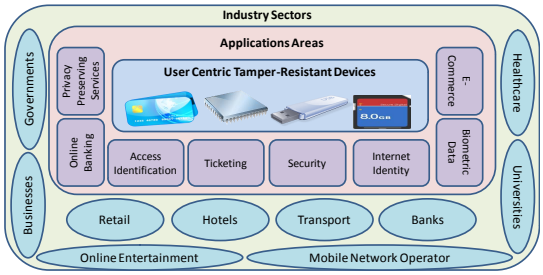

Figure 8. Illustration of UCTD form factors, application areas, and industry sectors

B. Trusted Cloud Computing

A generic logical architecture of cloud computing is illustrated in Figure 9, except for the trusted platform hardware, trusted platform OS and virtualised trusted agents. The hardware layer consists of processing and storage hardware. The processing hardware supports the execution environment (i.e. CPU, RAM, networking etc.), and storage hardware provides a data store. The processing platform Operating System (OS) manages communication between the cloud orchestrator and the processing hardware. Similarly, the storage platform OS facilitates communication between the cloud orchestrator and the storage hardware. The cloud orchestrator manages the resource allocations, networking and storage associated with individual virtual machines. Virtual machines are instances of clients' desired operating systems running in a virtualised environment to support their business requirements.

For hard trust, additional components are included in the generic architecture of cloud computing, thus providing a hardware-based trust assurance framework. These components are trusted platform hardware (e.g. TPM chip) and associated platforms (e.g. TPM functionalities). These trusted hardware and platforms are depicted as isolated segments of the cloud architecture, where in actual deployment they would be part of the processing and storage architectures. Each virtual machine

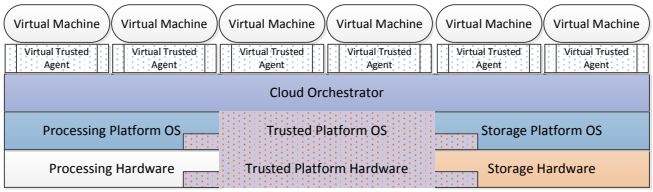

Figure 9. Trusted Computing Architecture for Cloud Computing

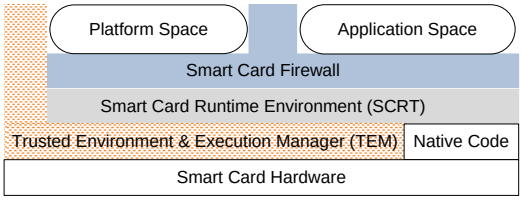

Figure 10. Smart Card Architecture in with TEM

can have a dedicated virtual trusted agent [47]. These virtual trusted agents can be built on the roots of trust provided by the underlying trusted platform (OS and hardware). This architecture can also support hybrid trust, where reputation-, context-, and content-based trust mechanisms can build on top of the trusted platform and virtual trusted agents.

C. Trusted Computing for Embedded Devices

For embedded devices, especially smart cards, a trusted entity known as the Trusted Execution & Environment Manager (TEM) has been proposed [32].

Such an entity would play a crucial role in establishing the trustworthiness of smart cards [48], platform assurance [49], smart card firewall mechanisms [50], trusted execution environments and smart card content backup/restoration mechanisms [51].

From the point of view of different stake-holders in various embedded devices (e.g. smart card), a TEM provides, at a minimum, the following services.

1) Confidence in Current State: Provides assurance and validation that the state of a smart card (software and hardware) is as secure as it was at the time of CC evaluation.
2) Trust in the Downloaded Application: Ensures that the application is downloaded and personalised in a secure and reliable fashion. Provides proof to the appropriate SP that there was no modification of the application during the download and installation process.
3) Secure State and Application Sharing: Provides assurance and validation services which an application can use to validate its current state, and verify the state of other application(s) with which it establishes application sharing.
4) Secure Execution: Provides a trusted execution environment which ensures that an application is executed in a trusted and secure environment.
5) Simulator Detection: Provides verification and validation mechanisms to ascertain both the existence of a smart card and that the item is not a smart card simulator (hardware genuineness [52]).

The overall architecture of a UCOM-based smart card is illustrated in Figure 10. The TEM is illustrated as a layer between the smart card hardware and the runtime environment. This illustration provides a semantic view of the architecture and does not imply that all communication between the runtime environment and the hardware goes through the TEM.

After analysing the TPM requirements [33], it is apparent that the basic building blocks in the hardware required to build a TPM chip are already available on smart cards. Therefore, most of the functionality of the TEM is implemented in software and it would not impose any additional hardware requirement on the host platform [32].

Figure 10 depicts native code and smart card hardware as complementary components of the TEM. This is because the TEM does not need separate hardware for its operations. It will utilise the existing services provided by the smart card hardware. To avoid duplicating the code, the TEM uses the native code implementation of cryptographic services like encryption/decryption, digital signature and random number generation.

VI. Conclusion and Future Research Directions

In this paper, we illustrated the difference between the notions of *digital trust* in different computing domains. The point of this discussion was to highlight that the definition of digital trust which is valid in one domain may not remain valid when applied to a different domain. Therefore, when discussing the notion of and potential models/frameworks for digital trust, the targeted domain should be properly scoped. In subsequent sections, we discussed digital trust for security and trust computing. In this discussion, we briefly elaborated on the traditional concepts of TPM and MTM and their relevance in today's changing technological landscape. Next, we put forward the *next* challenges that trusted computing faces, which include mobile platforms, cloud computing and cyber-physical systems. To satisfy the requirements of trusted computing and at the same time also accommodate these emerging technologies would require some fundamental modifications to the existing architecture for trusted computing. Finally, we described a wide range of proposals that built on traditional trusted computing but are extending it so they can satisfy the requirements of emerging technologies.

It can be argued that the concept of trusted computing is re-emerging and it might have a crucial role to play in future technologies. The challenge is how to create a trusted computing architecture that is flexible, extendable and scalable while providing the strongest possible security guarantees. How can we design a unified trusted computing architecture in a manner that can support security, privacy and trust-evaluation requirements ranging from the very small (e.g. embedded devices) to the very large (e.g. High Performance Computing)? These are important questions which should be considered by all in the trust management research and practice community.

REFERENCES

[1] J. Yao, J. Zhang, S. Chen, C. Wang, D. Levy, and Q. Liu, "A Mobile Cloud with Trusted Data Provenance Services for Bioinformatics Research," in *Data Provenance and Data Management in eScience*. Springer, 2013, pp. 109–128.

[2] A. Martin, J. Lyle, and C. Namilkuo, "Provenance as a Security Control," *TaPP. USENIX*, 2012.

[3] *ISO/IEC 11889-1: Information Technology - Trusted Platform Module - Part 1: Overview*, Online, ISO Standard 11 889-1, May 2009.

[4] *TPM Main: Part 1 Design Principles*, Online, Trusted Computing Group (TCG) Specification 1.2, Rev. 116, March 2011.

[5] "ARM Security Technology: Building a Secure System using TrustZone Technology," ARM, White Paper PRD29-GENC-009492C, 2009.

[6] "M-Shield Mobile Security Technology: Making Wireless Secure," Texas Instruments, Whilte Paper, February 2008.

[7] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing," *Dependable and Secure Computing, IEEE Transactions on*, vol. 1, no. 1, pp. 11–33, 2004.

[8] R. K. L. Ko, P. Jagadpramana, M. Mowbray, S. Pearson, M. Kirchberg, Q. Liang, and B. S. Lee, "Trustcloud: A framework for Accountability and Trust in Cloud Computing," in *Services (SERVICES), 2011 IEEE World Congress on*. IEEE, 2011, pp. 584–588.

[9] H. Kopetz, "Internet of Things," in *Real-Time Systems*, ser. Real-Time Systems Series. Springer US, 2011, pp. 307–323.

[10] E. A. Lee, "Cyber Physical Systems: Design challenges," in *Object Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEE International Symposium on*. IEEE, 2008, pp. 363–369.

[11] *Common Criteria for Information Technology Security Evaluation*, Common Criteria Std. Version 3.1, August 2006.

[12] T. Berners-Lee, J. Hendler, O. Lassila *et al.*, "The Semantic Web," *Scientific american*, vol. 284, no. 5, pp. 28–37, 2001.

[13] J. Golbeck, B. Parsia, and J. Hendler, "Trust Networks on the Semantic Web," in *Cooperative information agents VII*. Springer, 2003.

[14] T. Berners-Lee, W. Hall, J. A. Hendler, K. O'Hara, N. Shadbolt, and D. J. Weitzner, "A framework for web science," *Foundations and Trends in Web Science*, vol. 1, no. 1, pp. 1–130, 2006.

[15] T. Berners-Lee, "Semantic Web on XML," Presentation at XML,, 2000.

[16] C. Bizer and R. Oldakowski, "Using Context- and Content-based Trust Policies on the Semantic Web," in *Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers &Amp; Posters*, ser. WWW Alt. '04. New York, NY, USA: ACM, 2004, pp. 228–229.

[17] M. Richardson, R. Agrawal, and P. Domingos, "Trust Management for the Semantic Web," in *The Semantic Web-ISWC 2003*. Springer, 2003.

[18] S. Toivonen and G. Denker, "The Impact of Context on the Trustworthiness of Communication: An Ontological Approach," in *ISWC Workshop on Trust, Security, and Reputation on the Semantic Web*, vol. 127, 2004.

[19] I. Jacobi, L. Kagal, and A. Khandelwal, "Rule-based Trust Assessment on the Semantic Web," in *Rule-Based Reasoning, Programming, and Applications*. Springer, 2011, pp. 227–241.

[20] P. Buneman and Susan, "Data Provenance - the Foundation of Data Quality," September 2010.

[21] O. Q. Zhang, M. Kirchberg, R. K. L. Ko, and B. S. Lee, "How to track your data: The case for cloud computing provenance," in *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*. IEEE, 2011, pp. 446–453.

[22] R. Bose and J. Frew, "Lineage Retrieval for Scientific Data Processing: A Survey," *ACM Comput. Surv.*, vol. 37, no. 1, pp. 1–28, Mar. 2005.

[23] "EU PROVENANCE Project: An Open Provenance Architecture for Distributed Applications," in *Agent Technology and e-Health*, ser. Whitestein Series in Software Agent Technologies and Autonomic Computing, 2008, pp. 45–63.

[24] C. Dai, D. Lin, E. Bertino, and M. Kantarcioglu, "An Approach to Evaluate Data Trustworthiness Based on Data Provenance." in *Secure Data Management*, ser. Lecture Notes in Computer Science, W. Jonker and M. Petkovic, Eds., vol. 5159. Springer, 2008, pp. 82–98.

[25] I. Y. Jung and H. Y. Yeom, "Provenance Security Guarantee from Origin up to now in the e-Science Environment," *Journal of Systems Architecture*, vol. 57, no. 4, pp. 425–440, 2011.

[26] "Secure Data Management in Trusted Computing," in *Cryptographic Hardware and Embedded Systems - CHES 2005*, ser. Lecture Notes in Computer Science, J. Rao and B. Sunar, Eds. Springer, 2005.

[27] W. Zhou, Q. Fei, A. Narayan, A. Haeberlen, B. T. Loo, and M. Sherr, "Secure Network Provenance." in *SOSP*, T. Wobber and P. Druschel, Eds. ACM, 2011, pp. 295–310.

[28] R. N. Akram, K. Markantonakis, and K. Mayes, "An Introduction to the Trusted Platform Module and Mobile Trusted Module," in *Secure Smart Embedded Devices, Platforms and Applications*. Springer, NY, 2014.

[29] R. Zhou and K. Hwang, "PowerTrust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 4, pp. 460–473, Apr. 2007.

[30] K. Hwang and D. Li, "Trusted Cloud Computing with Secure Resources and Data Coloring," *IEEE Internet Computing*, vol. 14, pp. 14–22, 2010.

[31] O. Hartig and J. Zhao, "Using Web Data Provenance for Quality Assessment." in *SWPM*, ser. CEUR Workshop Proceedings, J. Freire, P. Missier, and S. S. Sahoo, Eds., vol. 526. CEUR-WS.org, 2009.

[32] R. N. Akram, K. Markantonakis, and K. Mayes, "Trusted Platform Module for Smart Cards," in *6th IFIP Intl Conference on New Technologies, Mobility and Security*, O. Alfandi, Ed. IEEE CS, March 2014.

[33] "Trusted Computing Group, TCG Specification Architecture Overview," TCG, Beaverton, Oregon, USA, revision 1.4, August 2007.

[34] *TCG Mobile Trusted Module Specification*, Online, Trusted Computing Group (TCG) Specification 1.0, Rev. 6, June 2008.

[35] G. Suh, C. O'Donnell, and S. Devadas, "Aegis: A Single-Chip Secure Processor," *Design Test of Computers*, December 2007.

[36] "Mobile NFC Services," GSM Association, White Paper Ver.1.0, 2007.

[37] *EMV 4.2*, Online, EMVCo Specification 4.2, May 2008.

[38] R. K. L. Ko, "Data Accountability in Cloud Systems," in *Security, Privacy and Trust in Cloud Systems*. Springer, 2014, pp. 211–238.

[39] D. Zissis and D. Lekkas, "Addressing Cloud Computing Security Issues," *Future Gener. Comput. Syst.*, vol. 28, no. 3, pp. 583–592, Mar. 2012.

[40] R. K. L. Ko, B. S. Lee, and S. Pearson, "Towards achieving accountability, auditability and trust in cloud computing," in *Advances in Computing and Communications*. Springer, 2011, pp. 432–444.

[41] C. H. Suen, R. K. L. Ko, Y. S. Tan, P. Jagadpramana, and B. S. Lee, "S2logger: End-to-End Data Tracking Mechanism for Cloud Data Provenance," in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE Intl. Conference on*. IEEE, 2013.

[42] O. Q. Zhang, R. K. L. Ko, M. Kirchberg, C. H. Suen, P. Jagadpramana, and B. S. Lee, "How to Track your Data: Rule-based Data Provenance Tracing Algorithms," in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*. IEEE, 2012, pp. 1429–1437.

[43] E. A. Lee, "CPS Foundations," in *Proceedings of the 47th Design Automation Conference*, ser. DAC '10. NY, USA: ACM, 2010.

[44] S. Seeber, A. Sehgal, B. Stelte, G. D. Rodosek, and J. Schönwälder, "Towards a Trust Computing Architecture for RPL in Cyber Physical Systems," in *9th International Conference on Network and Service Management, Zurich, Switzerland, October, 2013*, 2013, pp. 134–137.

[45] W. Rankl and W. Effing, *Smart Card Handbook*. New York, NY, USA: John Wiley & Sons, Inc., 2003.

[46] R. N. Akram, K. Markantonakis, and K. Mayes, "User Centric Security Model for Tamper-Resistant Devices," in *the 8th IEEE International Conference on e-Business Engineering (ICEBE 2011)*, J. Li and J.-Y. Chung, Eds. Beijing, China: IEEE Computer Science, October 2011.

[47] R. Perez, R. Sailer, L. van Doorn *et al.*, "vTPM: Virtualizing the Trusted Platform Module," in *15th Conf. on USENIX Security Symposium*, 2006.

[48] R. N. Akram, K. Markantonakis, and K. Mayes, "A Dynamic and Ubiquitous Smart Card Security Assurance and Validation Mechanism," in *25th IFIP Intl. Information Security Conference*, K. Rannenberg and V. Varadharajan, Eds. Australia: Springer, September 2010, pp. 161–171.

[49] ——, "Simulator Problem in User Centric Smart Card Ownership Model," in *6th IEEE/IFIP International Symposium on Trusted Computing and Communications (TrustCom-10)*, H. Y. Tang and X. Fu, Eds. HongKong, China: IEEE Computer Society, December 2010.

[50] ——, "Firewall Mechanism in a User Centric Smart Card Ownership Model," in *Smart Card Research and Advanced Application, 9th IFIP WG 8.8/11.2 International Conference*, ser. LNCS, D. Gollmann, J.-L. Lanet, and J. Iguchi-Cartigny, Eds., Passau, Germany: Springer, April 2010.

[51] ——, "Recovering from Lost Digital Wallet," in *The 4th IEEE International Symposium on Trust, Security, and Privacy for Emerging Applications (TSP-13)*, F. G. M. Y. Xiang and S. Ruj, Eds. Zhangjiajie, China: IEEE CS, November 2013.

[52] R. Kennell and L. H. Jamieson, "Establishing the Genuinity of Remote Computer Systems," in *Proceedings of the 12th conference on USENIX Security Symposium*. CA, USA: USENIX Association, 2003.