Modelling Dynamic Trust with Property Based Attestation in Trusted Platforms

Aarthi Nagarajan, Vijay Varadharajan

Modelling Dynamic Trust with Property Based Attestation in Trusted Platforms

Aarthi Nagarajan and Vijay Varadharajan

Macquarie University, Sydney, Australia
{aarthi,vijay}@ics.mq.edu.au

Abstract. Binary attestation in trusted computing provides the ability to reason about the state of a platform using integrity measurements. Property based attestation, an extension of binary attestation enables more meaningful attestation by abstracting low level binary values to high level security properties or functions of platforms. We believe that despite having trusted processes for integrity measurement, binary and property based attestation may still lead to ambiguities. These ambiguities may reduce the overall trust that can be placed on the measurements and properties that are attested by a platform. To address this issue, we propose TESM: a Trust Enhanced Security Model for trusted computing platforms. The overall aim of the model is to reduce the ambiguities and thereby enable better reasoning of properties that are satisfied by a platform with improved clarity.

1 Introduction

Trusted computing, standardised by the Trusted Computing Group (TCG) [1] provides techniques for achieving security using hardware in computing platforms. The core of the trusted computing technology is the Trusted Platform Module (TPM) chip that enables special functions in the platform. These functions include platform authentication that is used to ensure that the host platform is identifiable and genuine, secure storage for data and secrets, and platform attestation. Attestation, perhaps the key function of a trusted platform [1] provides the ability to reason about the state of a trusted platform in the form of hash measurements. A trusted platform consists of special measurement processes that measure every component installed on the platform at the time of boot and securely stores the measurements in the TPM chip. These measurements can then be reported to a third party who wishes to learn about the platform state. Based on the reported measurements, the third party may make judgements if the platform is in an acceptable and trustworthy state.

Recently, many researchers have proposed that it is more useful to reason about the state of a platform based on the security properties of the platform rather than plain hash measurement values [2, 3]. Several reasons for this have been put forth [2–4]. For example binary measurements change each time a component is updated and it is difficult to keep a record of all possible correct measurements while properties are more stable and do not change often

for trivial updates. To address this issue, an alternate form of attestation called property based attestation has been proposed. Property attestation leverages binary attestation to abstract low level binary hash values to high level security properties of platforms. The main aim of property attestation is to be able to prove that the availability of a certain hash measurement guarantees the availability of a certain security property. Several techniques for property attestation have been proposed recently and a comparison of these techniques can be found in [4]. In this paper, we adopt the certification based property attestation mechanism proposed in [3]. In this approach, a property certification authority (CA) evaluates the properties satisfied by a platform (or platform component) and issues a mapping between the expected hash of the platform to the properties satisfied in the form of a property certificate. If a platform measures up to the expected hash, then using the platform certificate it can prove that it satisfies the properties that are attested by the CA.

In this paper, we focus on the reliability of property based attestation. We believe that given the nature of the property attestation mechanism, certain ambiguities are introduced, which raises some fundamental questions on trusting the properties attested. We propose TESM: A Trust Enhanced Secure Model for trusted computing platforms. The overall aim of the model is to reduce ambiguities that arise in property based attestation; it takes into account the uncertainties and help reason about the properties of a system with better confidence. This is the important contribution of this paper. The rest of the paper is organised as follows. Section 2 outlines the motivation for the TESM model. Section 3 gives a basic introduction on subjective logic, which is used in the development of our trust model. Section 4 explains the Trust Enhanced Security Model (TESM) in detail. Section 5 describes how the trust model is being used in authorisation evaluation and we demonstrate this using an example scenario. Concluding remarks are given in section 6. A full description of the proposed model along with the architecture and implementation can be found in [5].

2 Motivation

The main aim of property based attestation is to abstract out binary measurements to more meaningful properties of systems. Once an attestation requester is able to reason about the properties of an attesting platform in a trustworthy manner, then these properties can be used in various security decision making processes. Currently, trust on property based attestation is derived from trust on binary attestation process which in turn is dependent on the trust on the measurement processes and the TPM that stores these measurements.

A fundamental question that arises then is - given that the process of property based attestation is significantly different from binary attestation and that the chain of trust is extended, how much can an Attestation Requester (AR) trust the properties that are presented by an Attesting Platform (AP)? In other words, when AP reports its system state with a set of properties to AR, how certain can AR be that these reported values are true and that AP actually satisfies

these claims. We belive that given the nature of property based attestation, uncertainties are introduced in the attestation process. This reduces trust on the property attestation process and leads to situations where AR cannot be completely certain if AP truly satisfies the properties presented to it. The reasons for such uncertainties in property based attestation have been listed below.

- In binary attestation, when AR requests AP for an attestation report, AR receives the measured values, reference values and the measurement list in response. These measurements that indicate the state of the components at the time of boot and not at the time of challenge. Today's systems are highly dynamic in nature and system components are constantly upgraded with updates from manufacturers. Furthermore, systems are also reasonably stable and they can go on without being rebooted for a very long time. This combination of a dynamic system that is not rebooted often means that values measured at boot time do not necessarily represent the state of the system at the time of attestation. This design admits potential for time-of-check time-of-use vulnerabilities: values reflect the state of the system when it was measured and not when it is reported. This makes the attestation report less useful. As the time between boot and attestation increases, AR is uncertain about how much it can trust the attestation report.
- In property attestation, AP proves that it satisfies a required set of properties using binary measurements and corresponding property certificates. These property certificates are issued by third party property certification authorities (CA). The process of property assessment and property certification by a CA does not happen on the run but much in advance before an attestation challenge is issued to AP. Also, property certificates are generated for each standalone component and not for the AP system as a whole. The reason being, with respect to security, it is easier to verify properties of individual components(which are smaller in size) than attempting to verify one large monolithic system.

  It is also possible that the environment under which the component is verified by CA could be different to the environment in which the component is measured during attestation. For example, a CA may install a component in its own system, verify it and certify it. When the component is later installed on a trusted platform, the component might not satisfy the property anymore because the state of the attesting platform is different from the state in which it was evaluated and certified. In some sense, this leads to the age old problem of secure composition of systems. Researchers have spent almost three decades trying to understand the composition of security properties in systems. The goal of secure composition process is to ensure that a composed system preserves the security properties of the individual constituent components. Two components may be individually evaluated for a property and certified. However, when the components are integrated in the same platform, it is often difficult to guarantee that they will preserve their properties under the influence of each other or other components that are already installed in the platform. The effects of composition might not even be reflected in the measurement values of the component.

This means that, even though the measured value and the certified value (in the property certificate) match, a property may still not be 'actually' satisfied.

– Like in any system that involves third party certification authorities, trust on the property certification authority is subjective and can vary depending on the context. The trustworthiness on the properties depends on the trustworthiness of the CA and verification mechanisms used by the CA to evaluate a component for a property. Therefore, trust on property attestation and the properties certified are directly dependent on AR's trust on the CA that certifies that property. If AR does not trust the honesty and the competency of the CA to verify and attest properties, AR may not trust the property certificates certified by that CA. Also, AR may trust a CA to certify one type of property but not other types. In some cases, AR may not even know if the CA is trustworthy or not. Such information must also be taken into consideration at the time of attestation verification.

In the next section, we take these uncertainties into consideration and design an automated trust model for property based attestation. The main aim of the model is provide a way of determining whether a platform can be trusted to satisfy a property given these uncertainties and how they can be factored into security decision making such as authorisation evaluation.

3 Context of the Model

This model is set in the context of subjective logic based belief modelling. Subjective logic proposed by Jøsang [6] is used to model trust that include uncertain outcomes. In this logic, trust is represented using an opinion metric which is denoted as ω where $\omega = (b, d, u)$ and b, d and u are belief, disbelief and uncertainty respectively. Values of $b, d, u \in [0, 1]$ and $b + d + u = 1$. Our model is based on subjective logic and some operations in this logic that are relevant to this paper are given below. We refer the reader to [6] for more details on subjective logic.

Evidence to Opinion mapping - Let *pos*, *neg*, *unc* denote the total number of positive, negative and uncertain experiences of A on B regarding the property x. Then A's opinion about property x in B is given by ${}^{A}\omega_B^x$ where ${}^{A}b_B^x$ is A's belief on B about x, ${}^{A}d_B^x$ is A's disbelief on B about x and ${}^{A}u_B^x$ is A's ignorance on B about x.

$$
\begin{aligned}
{}^{A}\omega_B^x &= {}^{A}b_B^x, {}^{A}d_B^x, {}^{A}u_B^x \\
{}^{A}b_B^x &= {}^{A}pos_B^x/({}^{A}pos_B^x + {}^{A}neg_B^x + {}^{A}unc_B^x) \\
{}^{A}d_B^x &= {}^{A}neg_B^x/({}^{A}pos_B^x + {}^{A}neg_B^x + {}^{A}unc_B^x) \\
{}^{A}u_B^x &= {}^{A}unc_B^x/({}^{A}pos_B^x + {}^{A}neg_B^x + {}^{A}unc_B^x)
\end{aligned}
$$

Conjunction of Opinions - Let A define two opinions ${}^{A}\omega_B^x$ and ${}^{A}\omega_B^y$ about two different properties x and y in the same platform B. Then ${}^{A}\omega_B^{x,y}$ is called the

conjunction ($\odot$) of the opinions ${}^{A}\omega_B^x$ and ${}^{A}\omega_B^y$ representing A's opinion about both x and y in B.

$$
\begin{aligned}
{}^{A}\omega_B^{x,y} &= {}^{A}\omega_B^x \odot {}^{A}\omega_B^y \\
{}^{A}b_B^{x,y} = {}^{A}b_B^x.{}^{A}b_B^y,\ {}^{A}d_B^{x,y} &= {}^{A}d_B^x + {}^{A}d_B^y - {}^{A}d_B^x.{}^{A}d_B^y \\
{}^{A}u_B^{x,y} &= {}^{A}b_B^x.{}^{A}u_B^y + {}^{A}u_B^x.{}^{A}b_B^y + {}^{A}u_B^x.{}^{A}u_B^y
\end{aligned}
$$

Consensus of Opinions - If A forms an opinion ${}^{A}\omega_B^x$ on B about the property x and C forms another opinion ${}^{C}\omega_B^x$ on B about the same property x, then the consensus ($\oplus$) of the two opinions is equivalent to the opinion ${}^{A,C}\omega_B^x$ formed on B about x by an imaginary system that represents both A and C.

$$
\begin{aligned}
{}^{A,C}\omega_B &= {}^{A}\omega_B^x \oplus {}^{C}\omega_B^x \\
{}^{A,C}b_B^x &= ({}^{A}b_B^x.{}^{C}u_B^x + {}^{C}b_B^x.{}^{A}u_B^x) \\
{}^{A,C}d_B^x = ({}^{A}d_B^x.{}^{C}u_B^x + {}^{C}d_B^x.{}^{A}u_B^x),\ {}^{A,C}u_B^x &= ({}^{A}u_B^x.{}^{C}u_B^x)
\end{aligned}
$$

Discounting Opinions - If A has an opinion on B and if B has an opinion on C, then A's opinion about C is computed by discounting B's opinion about C with A's opinion about B. Let ${}^{A}\omega_B = ({}^{A}b_B, {}^{A}d_B, {}^{A}u_B)$ and ${}^{B}\omega_C = ({}^{B}b_C, {}^{B}d_C, {}^{U}u_C)$, then ${}^{A}\omega_C$ gives the discounted opinion ($\otimes$) of ${}^{A}\omega_B$ and ${}^{B}\omega_C$.

$$
\begin{aligned}
{}^{A}\omega_X &= {}^{A}\omega_B \otimes {}^{B}\omega_C \\
{}^{AB}b_C = {}^{A}b_B.{}^{B}b_C,\ {}^{AB}d_C &= {}^{A}b_B.{}^{B}d_C \\
{}^{AB}u_C &= {}^{A}d_B + {}^{A}u_B + {}^{A}b_B.{}^{B}u_C
\end{aligned}
$$

4 Trust Enhanced Security Model

In this section, we present the formalisation of our automated trust model (ATM) for property attestation. The trust model ATM for a Trusted Platform TP can be defined as $ATM = (E, TR, OP)$. E represents the set of entities that share one or more trust relationships, TR is the set of trust relationships between the entities and OP is the set of operations for the management of trust relationships. We now define each entity below.

4.1 Entities of the trust model - *E*

The entities of the trust model share one or more trust relationships with each other. The trust model includes three different entities. First, there is the Attestation Requester (AR). AR is the entity that requests a trusted platform to attest to a set of of properties. The second entity is the Attesting Platform (AP). AP is the trusted platform that attests its state to AR. The third entity is the Certification Authority (CA). CA is the trusted party that issues expected measurement certificates and property certificates for components that are installed on AP.

4.2 Trust Relationship - TR

TR defines the trust relationship that is shared between two entities for a given property under a given set of conditions. Though trust can be defined in many ways, in the context of this model, our definition is similar to the notion expressed in [7] where trust is described as the firm belief in the competence of an entity to act dependably, reliably and securely within a specific context. Based on this, we have the following definitions.

Definition 1. Property Trust - *Property trust is the belief that a component in AP will satisfy a given property that has been certified for that component.*

Definition 2. Certification Trust - *Certification trust is the belief on the honesty and competency of a certification authority to certify a given property of a component.*

Definition 3. Trust Relationship - *A trust relationship TR is defined as* $TR = (A, B, C, P, K, \Theta, M, pos, neg, unc)$.

The tuple states that an entity A trusts an entity B for a component C to satisfy the property P with trust class K at a given time Θ with experience held in pos, neg, unc and opinion held in M. Entities A and $B \in E$; C is a member of $\{C\}$, a finite set of all components in B; P is a member of $\{P\}$, a finite set of all properties satisfied by C; K is a member of {satisfaction, certification} trust classes; Θ is the time at which experience values pos, neg, unc were last updated for this trust relationship i.e last update for this $\{TR\}$ occurred at time Θ; M is the evidence mapping operation on this trust relationship, which is presented as an opinion as defined in Definition 4; pos, neg, unc represent the total number of positive, negative and uncertain experiences respectively associated with this trust relationship.

4.3 Trust Management Operations - OP

This section outlines the different operations of the trust management system. The three main operations include evidence collection, opinion evaluation and opinion comparison. Each operation and its sub-operations are described below.

Evidence collection Evidence collection is the process by which AR records the outcome of its experience with AP for a given property. Evidence collection is divided into two parts, evidence collection from past and present experience. The first part represents the collection of evidence based on past experiences that have occurred prior to the time of authorisation θ. Experience is recorded on how well a platform satisfied a property in the past. Evidence collection on property satisfaction is still susceptible to ambiguities. Therefore, one must ensure that correct evidence is collected without uncertainty. For this purpose, table 1 is used for evidence collection and the mechanism is described below. In the second part of the evidence collection mechanism, property presented at the time of authorisation is translated into opinions. This is explained in the latter part of this section.

Evidence collection from past experiences - First, we describe the columns of the Table 1 below.

(i) Property Outcome - The property outcome column records if a property is satisfied by a platform or not. $p_s = 1$ indicates that a property is satisfied and $p_s = 0$ indicates that a property is not satisfied. Please note that this is the actual satisfaction of a property and not the property certificate validation outcome.
(ii) Events - The events column indicates if certain events have occurred in AP. A 'H-Event' can be considered as any change in AP that has occurred after the measurement time of a component and before the time of attestation report. A Pr-Event is an event that occurs in AP after property evaluation and certification but before property report. We combine both Pr-Events and H-Events events together as Pr/H Events. The value e = 0 indicates no events have occurred and the value e = 1 indicates one or more such events have occurred in AP. It must be noted that the occurrence of these events are not reflected in the attestation report as they occur after the time of measurement.
(iii) CA History - This represents the opinion about the honesty and/or ability of the CA to attest properties of a system in a correct manner.
(iv) Hash-History - This represents the opinion of AR on the validity of the hash measurement of a component in AP based on AR's past experiences with AP i.e how well a correct and current measurement was reported at the time of attestation.
(v) Experience recorded - This determines the experience recorded by AR about the satisfaction of a given property in AP. The main aim of the evidence collection operator is to populate this value given the other values in the table.

In table 1, we assume that all past experience outcomes recorded are absolute. AR either has complete belief, complete disbelief or complete uncertainty about a property. Correspondingly, belief (b), disbelief (d) and uncertainty (u) of hash H and certification authority CA are quantified as 0 or 1. Here, b(ca) = 1, d(ca) = 1 and u(ca) = 1 represent total belief, total disbelief and total uncertainty on the CA. Likewise, b(ca) = 0, d(ca) = 0 and u(ca) = 0 represent no belief, no disbelief and no uncertainty about CA respectively. Similarly, belief b(h) = 1, disbelief d(h)=1 and uncertainty u(h)=1 represent total belief, total disbelief and total uncertainty about the hash measurements of AP and b(h) = 0, d(h)=0 and u(h)=0 represent no belief, no disbelief and no uncertainty respectively. Table 1 has 4 categories. Each category defines a different condition in which experience is recorded.

(i) Category 1: An experience is recorded in the absence of CA history or hash history information. Here, when a property is satisfied, it is marked as a positive experience with respect to CA, property P and hash H. This is irrespective of whether events have occurred (that is, e=1) or events have not occurred in the

Table 1. Evidence collection I

Cat	Pr-Outcome	Pr/H-Events	CA-History	Hash-History	Experience-recorded
1	$p_s = 1$	e = 0/1	not available	not available	pos(ca,p,h)
	$p_s = 0$	e = 0/1			neg(p),unc(ca,h)
2	$p_s = 1$	e = 0	any b(ca),d(ca),u(ca)	any b(h),d(h),u(h)	pos(ca,p,h)
	$p_s = 1$	e = 1	any b(ca),d(ca),u(ca)	any b(h),d(h),u(h)	pos(ca,p,h)
3	$p_s = 0$	e = 0	b(ca) = 1	b(h) = 1	neg(p)
	$p_s = 0$	e = 0	b(ca) = 1	d(h) = 1	neg(p), neg(h)
	$p_s = 0$	e = 0	b(ca) = 1	u(h) = 1	neg(p), unc(h)
	$p_s = 0$	e = 0	d(ca) = 1	d(h) = 1	neg(p),unc(ca,h)
	$p_s = 0$	e = 0	d(ca) = 1	b(h) = 1	neg(ca,p)
	$p_s = 0$	e = 0	d(ca) = 1	u(h) = 1	neg(p),unc(ca,h)
	$p_s = 0$	e = 0	u(ca) = 1	u(h) = 1	unc(ca,h),neg(p)
	$p_s = 0$	e = 0	u(ca) = 1	b(h) = 1	unc(ca),neg(p)
	$p_s = 0$	e = 0	u(ca) = 1	d(h) = 1	unc(ca,h),neg(p)
4	$p_s = 0$	e = 1	b(ca) = 1	b(h) = 1	neg(p)
	$p_s = 0$	e = 1	b(ca) = 1	d(h) = 1	neg(p),unc(h)
	$p_s = 0$	e = 1	b(ca) = 1	u(h) = 1	neg(p),unc(h)
	$p_s = 0$	e = 1	d(ca) = 1	d(h) = 1	neg(p),unc(ca,h)
	$p_s = 0$	e = 1	d(ca) = 1	b(h) = 1	neg(p),unc(ca)
	$p_s = 0$	e = 1	d(ca) = 1	u(h) = 1	neg(p),unc(ca,h)
	$p_s = 0$	e = 1	u(ca) = 1	u(h) = 1	neg(p),unc(ca,h)
	$p_s = 0$	e = 1	u(ca) = 1	b(h) = 1	neg(p),unc(ca)
	$p_s = 0$	e = 1	u(ca) = 1	d(h) = 1	neg(p),unc(ca,h)

system. When a property is not satisfied, it is marked as a negative experience with respect to the property and uncertainty with respect to the CA and H. This is because, it is not possible to determine if H (hash being invalid) or CA (not certified correctly) contributed to the property being invalid.

(ii) Category 2: An experience is recorded when history information about CA and H is available but this history does not influence the outcome of the experience. When a property is satisfied, a positive experience pos(p) is recorded irrespective of the past experience with that platform. A satisfied property also increases belief in the hash and the certification authority and leads to a positive experience for both H and CA.

(iii) Category 3: In this category, an experience is recorded when a property is not satisfied and there have been no Pr/H-Events. In all these cases, because the property was not satisfied, outcome for P is marked as a negative experience. In order to record an experience for CA and H, we in turn use the history information associated with CA and H respectively. If b(ca) = 1 and b(h) = 1, then there is only a negative experience for P. However, if b(ca) = 1 and d(h) = 1, AR can ascertain based on its past experience of d(h)=1 that the most likely reason that the property failed is because the component's hash changed

after boot time measurement. This leads to a negative experience for H. (We have taken a stronger approach and have marked a negative experience for H. Alternatively, a more lenient approach may be taken and H may be marked as uncertain.) If b(ca)=1 and u(h)=1, then AR cannot determine if the property failed because of the component hash as it is itself uncertain about the past experiences of H. This leads to an uncertain experience for H. Alternatively, if AR totally disbelieves CA (d(ca) = 1) and totally believes H (b(h) = 1), this records a negative experience for CA as AR believes in H completely and will rule out H as a possible reason (again a more lenient approach may be taken and CA may be marked as uncertain, but we take a stronger approach). If d(ca) = 1 and u(h) or d(h) = 1, then the platform is not certain if the property failed because of the hash or CA. So it marks this as an uncertain experience for both H and CA. Similarly, if AR is uncertain about CA, u(ca)=1 and the property is not satisfied, then AR marks CA as uncertain again as it cannot be sure if the property was not really satisfied or if the CA had wrongly certified the property. Additionally, if AR does not have total belief in the hash of the component, i.e. if d(h)=1 or u(h)=1, then a possible bad hash measurement also adds to the existing uncertainty. Therefore, an uncertain experience unc(h) is marked along with unc(ca).

(iv) Category 4: In category 4, an experience is recorded when a property is not satisfied, when events have occurred, and when history of hash and CA are available. Here as well, when a property is not satisfied, it is always a negative experience with respect to P. In order to record values for CA and H, we use the history information of CA and H respectively. The main difference compared to category 3 is that the events introduce even more uncertainty. For instance, when there is total belief in CA and total disbelief in H, one cannot still record this as a negative experience with respect to H as in the previous case. There is uncertainty as to whether the events lead to the property being invalid or the hash value change after boot. Therefore, we mark an uncertain experience for H and not a negative experience. Similarly, when there is total belief in H and total disbelief in CA, one cannot record this as a negative experience for CA. It is not clear if the events lead to the property being invalid or if CA certified the property wrongly. Therefore, we mark an uncertain experience for CA and not a negative experience. All other experiences are recorded using similar judgement as in category 3 in the table.

Evidence collection from present experience - So far, we have discussed how AR collects evidence about platform AP for a given property based on the past behaviour. At the time of service request θ, AR presents property certificates to AP to vouch for the current state of the platform. We believe that this information must also be taken into account while computing the overall opinion of a platform. In order to take into account not only the past experiences but also the present state of the platform, we record the successful validation of a property certificate as a positive experience *pos* with respect to that property. If the property certificate is verified, then the opinion formed at the time of attestation is considered as (1,0,0) where belief is $\frac{1}{1}$, disbelief and uncertainty are

$\frac{0}{1}$ as obtained from the evidence to opinion mapping operator given in section 3. Similarly, if a property certificate is not verified, then the opinion becomes (0,1,0). There is no uncertainty here as a property certificate either validates or does not validate.

Trust Evaluation: Section 3 describes subjective logic as the main context of this trust model. Subjective logic uses special belief functions called opinions to represent trust. An opinion metric is given by $\omega = (b, d, u)$ where b represents belief, d represents disbelief and u represents uncertainty for a given trust relationship. We adopt this representation of trust as an opinion metric in our trust model.

Definition 4. ***Evidence Mapping***

The evidence mapping operator M (Definition 3) on a given trust relationship TR is used to represent the opinion of one entity on another entity. The main function of the mapping operator is to map the collected evidence in the form of positive, negative and uncertain experiences to an opinion value. This is achieved using the evidence to opinion mapping function of subjective logic as given in section 3.

Opinion Decay - Trust is dynamic in nature and tends to change with time. Over a given time frame, the value of trust in the beginning of the period is different from the value in the end even when there are no underlying factors that affect the value of trust directly. Just as we humans tend to forget things or associate less importance to events that have occurred in the past, we model systems also to associate less importance to events that have occurred in the past compared to more recent events. In other words, the system is modelled to gradually become non-decisive about trust (and distrust) as time progresses. The decay operator is a function that is used to represent this nature of trust. Equation 1 shows the decay function $\Psi_{k,\Delta}$ used to calculate new opinion ω_{new} after decay from an old opinion ω_{old}.

$$\omega_{new} = \Psi_{k,\Delta}[\ \omega_{old}] \tag{1}$$

Where $\Psi_{k,\Delta}$ is gives as

$$\begin{aligned} b_{new} &= b_{old}[1 - e^{-(k.\Delta)}] \\ d_{new} &= d_{old}[1 - e^{-(k.\Delta)}] \\ u_{new} &= u_{old} + [(b_{old} + d_{old}) - (b_{new} + d_{new})] \end{aligned}$$

(i) k is the rate of decay and $k > 0$ & $k \leq 1$. For example, if the rate of decay is 1 %, then k = 0.01, if rate of decay is 10 %, then $k = 0.1$ and if rate of decay is 100 %, $k = 1$.

(ii) Δ is the difference between the current time θ at which service is requested and the time at which opinion for that property was last updated. The value of Δ is chosen such that ω_{new} does not decay rapidly. For example, if $\Delta = 0$, then ω_{new} is same as ω_{old} and if $\Delta = \infty$ then ω_{new} tends to zero. In our model, we chose Δ as the number of years that have elapsed since the opinion was last updated. The minimum value of Δ is 0/365 (zero days) and the maximum value is 730/365 (2 years approximately). At $\Delta = 2$ and $k = 1$, b_{new} is 13 percent of b_{old}. This is the maximum decay possible for any opinion. Any time greater than 2 years is also assumed to be 2 years such that b_{new} decays at 100% to a maximum of $0.13(b_{old})$ and not more.

Total opinion on a property p_j of a component c_i - Let $({}^{A}pos_{B,sat(c_i,p_j)}, {}^{A}neg_{B,sat(c_i,p_j)}, {}^{A}unc_{B,sat(c_i,p_j)})$ represent the evidence associated with a trust relationship TR of platform A about platform B for the satisfaction of a property p_j of a component c_i. Based on the evidence collected, the evidence mapping function M is used to calculate the opinion for this TR. This defines the opinion of A about platform B for the satisfaction of property p_i of component c_i at time Θ and is given as

$${}^{A}\omega^{\Theta}_{B,sat(c_i,p_j)} = \{{}^{A}b^{\Theta}_{B,sat(c_i,p_j)}, {}^{A}d^{\Theta}_{B,sat(c_i,p_j)}, {}^{A}u^{\Theta}_{B,sat(c_i,p_j)}\}$$

Let ${}^{A}pos_{CA,cer(c_i,p_j)}, {}^{A}neg_{CA,cer(c_i,p_j)}, {}^{A}unc_{CA,cer(c_i,p_j)}$ represent the evidence associated with a Trust Relationship TR of a platform A about Certification Authority CA for the certification of the property p_j of a component c_i. Evidence mapping function M is used to calculate the opinion for this TR. This defines the opinion of platform A about CA for the certification of a property p_j of a component c_i at time Θ and is given as

$${}^{A}\omega^{\Theta}_{CA,cer(c_i,p_j)} = \{{}^{A}b^{\Theta}_{CA,cer(c_i,p_j)}, {}^{A}d^{\Theta}_{CA,cer(c_i,p_j)}, {}^{A}u^{\Theta}_{CA,cer(c_i,p_j)}\}$$

The total opinion on the property p_j of the component c_i is calculated by combining the satisfaction opinion (how well the property was satisfied) and certification opinion (how well the CA certified that property) of that property. Equation 2 gives the conjunction (section 3) of both these opinions. The opinions are decayed using equation 1. Θ gives the time at which experience was last updated.

$${}^{A}\omega^{\Theta}_{B,c_i,p_j} = \Psi_{k,\Delta}[{}^{A}\omega^{\Theta}_{B,sat(c_i,p_j)}] \odot \Psi_{k,\Delta}[{}^{A}\omega^{\Theta}_{CA,cer(c_i,p_j)}] \tag{2}$$

Direct Trust - Direct Trust is the belief one entity holds on another entity for a given context, based on its own past experiences with that entity. The direct trust on a component c_i and a property p_j is calculated by combining the total opinion formed at time of service request θ and the total opinion formed prior to service request at time $\theta - t$.

$${}^{A-dir}\omega_{B,c_i,p_j} = {}^{A}\omega^{\theta}_{B,c_i,p_j} \odot {}^{A}\omega^{\theta-t}_{B,c_i,p_j} \tag{3}$$

where, ${}^{A}\omega^{\theta}_{B,c_i,p_j}$ and ${}^{A}\omega^{\theta-t}_{B,c_i,p_j}$ are total opinions at times θ and $\theta-t$ respectively and $\theta, \theta-t \in \Theta$. The total opinions ${}^{A}\omega^{\theta}_{B,c_i,p_j}$ and ${}^{A}\omega^{\theta-t}_{B,c_i,p_j}$ are derived using equation 2.

$$ {}^{A}\omega^{\theta-t}_{B,c_i,p_j} = \Psi_{k,\Delta}[{}^{A}\omega^{\theta-t}_{B,sat(c_i,p_j)}] \tag{4} $$

$$ {}^{A}\omega^{\theta}_{B,c_i,p_j} = \Psi_{k,0}[{}^{A}\omega_{B,sat(c_i,p_j)}]^{\theta} \odot \Psi_{k,\Delta}[{}^{A}\omega^{\theta-t}_{CA,cer(c_i,p_j)}] \tag{5} $$

(i) In equation 3, the most recent opinion at θ is combined with all the previous experiences prior to time θ. Although the equation does not attach weightage to the opinions, clearly the opinion formed using the experience at time θ has more influence that any other individual experience prior to this time. If the opinion at time θ is (1,0,0), then the value of direct opinion is equal to the opinion at time $\theta-t$. An opinion (1,0,0) at θ is possible if the present experience is positive (evidence collection from present experience given in section 4.3) and the opinion of the privacy CA is (1,0,0). This is expected to be the usual case. If privacy CA is not completely trusted, then this reduces the value of the direct opinion dramatically due to the nature of the $\odot$ operator. The notion behind this is, if the CA is not trusted, then the certified property itself may not be trusted.

(ii) If a platform has had no direct experiences with respect to a property, ${}^{A}\omega^{\theta-t}_{B,sat(c_i,p_j)}$ is assumed as (1,0,0) in equation 3. This makes the direct opinion equal to opinion ${}^{A}\omega_{B,c_i,p_j}$ at θ.

(iii) Equation 5 is derived from equation 2. Here, the value of ${}^{A}\omega^{\theta}_{B,sat(c_i,p_j)}$ does not decay because it is computed based on the evidence recorded at the time of service request θ and the value of $\Delta = 0$. ${}^{A}\omega^{\theta}_{CA,cer(c_i,p_j)}$ represents the certification trust on CA that certifies the property p_i. When the past experience of this CA is unavailable, the value of ${}^{A}\omega^{\theta-t}_{CA,cer(c_i,p_j)}$ is assumed to be (1,0,0) which makes ${}^{A}\omega^{\theta}_{B,c_i,p_j} = \Psi_{k,0}[{}^{A}\omega^{\theta}_{B,sat(c_i,p_j)}]$

(iv) Equation 4 is derived from equation 2 where Θ is equal to $\theta - t$. Certification opinion on CA at time $\theta - t$ is removed from equation 4. This is because ${}^{A}\omega^{\theta-t}_{CA,cer(c_i,p_j)}$ is formed using a collection of certification outcomes for the property p_i. This property could have been certified by different certification authorities in the past. Therefore, it is not possible to attribute the certification opinion to any one single CA.

Recommended Trust - Recommended trust is the belief one entity holds on another entity for a given context, based on the recommendations obtained from its peer entities' past experiences. ${}^{A-rec}\omega_{B,c_i,p_j}$ represents the overall recommended opinion of A on B computed from the individual opinions of A's recommenders.

$$ {}^{A-rec}\omega_{B,c_i,p_j} = (I_{R_1} \otimes \Psi_{k,t}\ [{}^{R_1}\omega^{\theta-t}_{B,sat(c_i,p_j)}]) \oplus .. \tag{6} $$
$$ \oplus (I_{R_m} \otimes \Psi_{k,t}\ [{}^{R_m}\omega^{\theta-t}_{B,sat(c_i,p_j)}]) $$

(i) A decay function is applied to each recommended opinion to ensure that the value of the recommenders' opinion decrements with time.
(ii) For each decayed opinion of a recommender, an importance factor I is attached to the respective recommendations. The importance factor determines how much platform A values each recommender. The importance factors of $I_{R_1}..I_{R_m}$ are attached to the decayed opinions of recommenders $R_1..R_m$ respectively using the discounting operator given in section 3. $I_R = (wt, 1-wt, 0)$ where wt denotes the weight for a recommender R. The sum of the weights of all recommenders equals 1.
(iii) If there is more than one recommender, R_1 to R_m, then a consensus of every recommender's weighted decayed opinion is computed using the consensus operator $\oplus$ given in section 3.

Derived Trust - Derived trust is the belief one entity builds on another entity for a given context, based on other atomic trust relationships such as direct trust and recommended trust. Derived opinion for a property p_j of component c_i is computed by combining the direct and recommended opinions for that property.

$$^{A-der}\omega_{B,c_i,p_j} = {}^{A-dir}\omega_{B,c_i,p_j} \oplus {}^{A-rec}\omega_{B,c_i,p_j} \tag{7}$$

In equation 7, A computes derived opinion for property p_i of component c_i by combining its recommended opinion from equation 6 with its direct opinion from equation 3. If a recommended opinion is unavailable, then derived opinion is equal to the direct opinion. In the absence of direct opinion, derived opinion equals recommended opinion.

Derived Platform Trust - Derived platform trust is defined as the belief one platform holds on another platform for a given context based on the combined belief of the individual properties of that platform. Platform trust of A on B is computed by combining the derived opinions of all the properties in B.

$$^{A-der}\omega_B = I_{c_i,p_j} \otimes \Psi_{k,t}\,[{}^{A-der}\omega_{B,c_i,p_j}] \oplus I_{c_k,p_l} \otimes \Psi_{k,t}\,[{}^{A-der}\omega_{B,c_k,p_l}] \tag{8}$$

Assuming that platform B has two properties - property p_j of component c_i and property p_l of component c_k where $p_j, p_l \in P$ and $c_i, c_k \in C$, then the derived platform opinion $^{A-der}\omega_B$ equals the consensus of the derived opinions $^{A-der}\omega_{B,c_i,p_j}$ and $^{A-der}\omega_{B,c_k,p_l}$ on properties p_j and p_l respectively. The opinion are decayed using the decay operator given in equation 1. An importance factor I is attached to each opinion. This importance factor determines how much A values each property to contribute to the overall trust of the platform. Sum of the importance factors is equal to 1 and $I = (wt, 1-wt, 0)$ where wt denotes the weight for each property.

Trust Comparison: An opinion comparison operator $\geq_\omega$ that compares any two given opinions ω_1 and ω_2 is defined. Given two opinions ω_1 and ω_2, we define an opinion comparison operator $\geq_\omega$, whereby $\omega_1 \geq_\omega \omega_2$ holds if $b_1 > b_2$, $d_1 < d_2$ and $u_1 < u_2$. In such cases, we say that ω_1 is greater than the threshold presented by ω_2.

5 Authorisation Evaluation

We have previously defined a formal trust relationship TR. When a platform A (attestation requester) makes a request to another platform B (attestation provider) for some service, platform A must determine based on its existing trust relationship with platform B, if platform B will be allowed to access the service or not. Here, platform B presents a request to platform A with its measurement and property certificates that A requires in order to service the request. Using the property certificates and the trust relationship of A on B's properties, A can determine if it can trust platform B to really satisfy these properties.

An overall picture of the authorisation process can be given as follows. Initially, A computes its direct opinion of a property for platform B using the equation 3. If possible, A looks for recommenders that can provide recommendations for the properties satisfied by platform B. Each recommended opinion is decayed for the time elapsed since the last recommendation was recorded. It is possible that the decay time Δ is different for each recommender. Also note that A can define a different decay rate k and important rate I for each recommender. The final recommended opinion is then calculated using equation 6. Then the direct and recommended opinions are combined together using equation 7 to compute the derived opinion. Alternatively, A may compute B's overall platform trust using equation 8. For every service that is provided, A defines authorisation policies that include a threshold value ω_{th} as an opinion constant. A compares the derived opinion and threshold opinion using the comparison operator $\geq_{\omega}$. If the derived opinion is higher, then A services B's request.

A working architecture of the authorisation model has been implemented and several other design choices have also been made available. On the one hand, platform A may opt to make authorisation decisions using different trust groups; for example, with direct opinion alone, when recommendation trust is unavailable. Derived opinion is used as a default design choice. On the other hand, different authorisation parameters for soft trust may be defined. Opinion thresholds may be defined for an overall platform (${}^{A}\omega_{B}$) or for individual properties of components (${}^{A}\omega_{B,c_i,p_j}$). In order to derive an overall threshold for a platform, individual ${}^{A}\omega_{B,c_i,p_j}$ are combined together using the consensus operator (section 3). Before consensus, weights for each ${}^{A}\omega_{B,c_i,p_j}$ may also be applied using the discounting operator, similar to the application of I in recommended opinion calculation. When thresholds for individual c_i, p_i are used, comparison operator is applied to each threshold and ${}^{A}\omega_{B,c_i,p_j}$ pair, and the outcomes are ANDed together for a final decision. A combination of both overall platform opinion and opinion on individual properties of components is also possible. A full description of the architecture and policy scenarios can be found in [5].

5.1 Example Scenario

In this section, we present an example scenario for the trust model. Here, a large number of past experience record is available but the number of recommendations that can be gathered is limited in number. Let us take the example of an

online gaming system. In this system, the game provider must ensure that each of the participants is using the correct version of the gaming software that satisfies a required set of properties. Participants usually cheat the game provider by modifying the gaming software G to their advantage. By ensuring that a participant always plays with a software that is 'unmodified', the game provider can ascertain that all the participants are playing honestly and every participant has a fair chance to win. We assume that the game provider A has previous experiences with a participant X and has recorded all previous experience outcomes. The game provider is also able to obtain recommendations about X from two other game providers B and C that X has previously interacted with. Recommendations from both the recommenders are given equal weightage. The game provider is willing to allow X to participate if the derived trust on X for the given software is greater than a threshold opinion of (0.5,0.5,0). Time of authorisation is November 12 2009, 14:00 hrs. Opinions are decayed at the rate of 100%. The following trust relationships are available in the trust base.
(a) $(A, X, G, unmodified, sat, Oct\ 01\ 2009, 14:00:00, [0.789, 0.105, 0.105], 15, 2, 2)$
(b) $(A, X, G, unmodified, sat, Nov\ 12\ 2009, 14:00:00, [1, 0, 0], 1, 0, 0)$
(c) $(A, CA, G, unmodified, cert, Oct\ 01\ 2009, 14:00:00, [0.923, 0, 0.77], 12, 0, 1)$
(d) $(B, X, G, unmodified, sat, Oct\ 31\ 2009, 14:00:00, [0.166, 0.833, 0], 3, 15, 0)$
(e) $(C, X, G, unmodified, sat, Oct\ 03\ 2009, 14:00:00, [0.1, 0.90, 0], 2, 20, 0)$

Now we compute the opinions using these relationships.

(1) **Direct Trust**
- Opinion ${}^{A}\omega^{\theta-42/365}_{X,G,unmodified}$ that software G is unmodified in X based on experience updated 42 days prior request (from eqn 4) = (0.70, 0.09, 0.20)
- Opinion ${}^{A}\omega^{\theta}_{X,G,unmodified}$ that software G is unmodified in X based on experience recorded at the time of request (from eqn 5) = (0.82,0,0.17)
- Direct ${}^{A-dir}\omega_{X,G,unmodified}$ opinion that software G is unmodified in X (from eqn 3) = (0.57,0.09,0.32)

2) **Recommended Trust**
- Recommended opinion ${}^{B}\omega^{\theta-12/365}_{X,sat(G,unmodified)}$ of B based on evidence recorded 12 days prior request = (0.16,0.83,0)
- Recommended opinion ${}^{C}\omega^{\theta-40/365}_{X,sat(G,unmodified)}$ of C based on evidence recorded 40 days prior request = (0.09, 0.90, 0)
- Total recommended opinion ${}^{A-rec}\omega_{X,G,unmodified}$ after decay and with equal weights of 0.5 for both B and C (from eqn 6) = (0.08, 0.55, 0.36)

(3) **Derived Trust**
Derived opinion ${}^{A-der}\omega_{X,G,unmodified}$ of A that software G is unmodified in platform X (From eqn 7) = (0.42, 0.37, 0.20)

The derived opinion is compared against the threshold opinion of (0.5,0.5,0.0) using the comparison operator. The derived opinion is not greater than the threshold and the game player does not permit X to participate in the game.

One can see that although the direct opinion of (0.57,0.09,0.32) is greater than the threshold, the inclusion of the recommended opinions has yielded a different outcome with opinion (0.42,0.37,0.20) being less than the required threshold.

6 Conclusion

Property based attestation is an extension of the TCG attestation mechanism where binary hash measurements are abstracted to meaningful properties of systems. Recently, property based attestation has gained considerable interest in the research community as properties of systems are more persistent and do not change like hash measurements for trivial changes in system configuration. In this paper, we have proposed a Trust Enhanced Security Model (TESM) that models dynamic trust for binary and property based attestation in trusted platforms. We have shown that both binary attestation and property based attestation introduce uncertainties in the attestation mechanism and due to this an attestation requester is unable to reason about the trustworthiness of an attesting platform with absolute certainty. To address this issue, we have proposed a trust enhanced security model that derives trust from property certificates and social control mechanisms like past experiences and recommendations of how well a platform behaved in the past. We have described how evidence about a platform is collected and how opinions are formed using the collected evidence. Using these opinions, an attestation requester is better able to gauge how well a platform will behave in the future with reduced uncertainty. We believe such a model is useful to enhance the attestation process and this will enable reasoning the trust on attesting platforms with greater confidence.

Acknowledgements. We would like to thank the anonymous reviewers and the program committee for their valuable comments and suggestions on the paper.

References

1. Trusted Computing Group: TPM Main - Part 1 Design Principles, Version 1.2, Revision 103. (July 2007)
2. Poritz, J., Schunter, M., Herreweghen, E.V., Waidner, M.: Property attestation-Scalable and privacy-friendly security assessment of peer computers. Technical report, IBM Research (May 2004)
3. Sadeghi, A.R., Stüble, C.: Property-based attestation for computing platforms: Caring about properties, not mechanisms. In: NSPW '04: Proceedings of the 2004 Workshop on New Security Paradigms, USA, ACM (2004) 67–77
4. Nagarajan, A., Varadharajan, V., Gallery, E., Hitchens, M.: Property based attestation and trusted computing: Analysis and challenges. In: Third International Conference on Network and System Security, Gold Coast, Australia (October 2009)
5. Anonymous: Title Suppressed. PhD thesis (2010)
6. Jøsang, A.: A logic for uncertain probabilities. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. **9**(3) (2001) 279–311
7. Grandison, T., Sloman, M.: A survey of trust in internet applications. IEEE Communications Surveys and Tutorials **3**(4) (2000)