User-Centered Attestation for Layered and Decentralized Systems (Preprint)

Hagen Lauer
Monash University
hagen.lauer@monash.edu

Ahmad Salehi S.
Monash University
ahmad.salehishahraki@monash.edu

Carsten Rudolph
Monash University
carsten.rudolph@monash.edu

Surya Nepal
Data61 / CSIRO
surya.nepal@data61.csiro.au

ABSTRACT

Virtualization is omnipresent as the backbone of cloud, edge, and fog computing as well as *X-as-a-service* infrastructure. It continues to gain increased popularity even in edge or end-user and embedded devices. The need for standards and specifications for secure and trustworthy collaboration becomes a pressing issue. Trusted Computing is considered one of the pillars towards trustworthy systems both in terms of practical security mechanisms and supporting standards. This paper revisits the Trusted Computing tool-set and introduces its current application in virtualization scenarios. We discuss challenges related to translating the term *trust* between specifications for hardware modules such as the Trusted Platform Module (TPM) and applied specifications for operating systems, hypervisors, and virtual machines are — defining trust establishment becomes crucial for specifications extending *trust* beyond the TPM. We define User-centered attestation as a set of principles suitable for layered, decentralized systems along with a methodology for specifying and synthesizing such a trust establishment strategy.

1 INTRODUCTION

Cloud Computing has become ubiquitous in a plethora of applications ranging from education, finance, and smart home to healthcare, government, and military applications [2, 5, 17, 22]. However, the cloud-centered paradigm faces a major shift: the success of the Internet-of-Things causes the generation of the majority of data at the outer edges of a network [10, 22, 23]. *Edge computing* refers to the set of technologies allowing computations to be performed along the edges of the network, on downstream data on behalf of cloud services and upstream data on behalf of IoT services [10, 22]. *Fog computing* is closely related to the general concept of *edge computing*[5, 7, 27] with a strong focus on performing task in nearby, decentralized systems. For some tasks, this yields considerable feats such as lower latency and improved user-experience as well as resilience through redundancy for services. Virtualization, essential for concepts ranging from *Infrastructure-as-a-Service* to *Functions-as-a-Service* implemented by vendors like Microsoft, Google, and Amazon has fundamentally changed the way software and data is being handled and is the backbone of modern computing infrastructure, especially with an increase in service decentralization and dedicated, collaborating nodes[5, 17, 22].

The importance of trust and trust establishment strategies becomes apparent in decentralized systems with no immediately recognizable authorities. The rather ambiguous issue of trust and collaboration can be demonstrated using a very small, discrete example: Suppose a mathematician who is interested in number theory uses a computer with a program for factorizing numbers. The output that will be produced by that program is either the factorization of a given number or a statement that the given number is a prime. Now suppose that the same mathematician wishes to inspect a large number, too large to verify without the aid of the computer. The mathematician can have two possible expectations at this point: the given number is a prime number or not. Assuming there are strong reasons to believe that the number is a prime, the result of the program can either confirm this by telling that the number is a prime or give the factorization as evidence that intuition has in fact fooled the mathematician. The situation changes, however, if the mathematician has strong reasons to believe that the given number is not prime. Again, the computer can produce two possible outputs: the number is a prime or a factorization. If the output is a factorization, the mathematician can confirm the belief by recalculating the given number. However, if the computer comes back with the result that the number is a prime, contrary to strong reasons leading one to believe otherwise, why should the mathematician trust this result? [9] This example illustrates that even in completely discrete problems, the computation may not be worthwhile if it lacks convincing power w.r.t. the *quality* of the result. As possibly dated and oversimplified as it may seem, the issue raised here, instead of being remedied, is being amplified by modern efforts using more flexible, decentralized computing systems. As a practical set of standards-based technologies, Trusted Computing [11] can serve to supply evidence about a computing platform. The process of collecting, supplying, and appraising evidence, and ultimately a result, is referred to as Remote Attestation [6, 8, 15].

1.1 Contribution

This paper introduces current technological and standardization efforts towards trustworthy cloud computations. Implementing trustworthy virtualized systems currently requires the adoption of at least two standards for hardware and application level trust. We outline challenges and potential conflicts related to translating *trust* across such standards. We then review and evaluate current trust establishment methods and put them into perspective of decentralized systems. Finally, we propose user-centered attestation as a

candidate for layered, decentralized systems along with a methodology and strategy for specifying and synthesizing such an attestation system.

2 TRUSTED VIRTUALIZATION PLATFORM

The notion of a trusted visualized platform is coined by Trusted Computing Group's (TCG) companion architecture specification "Virtualized Trusted Platform Architecture" [24]. The TCG coined term and the architecture itself is rather vague but decisively extends the adjective *trusted*, which should cause curiosity since intuitively only the TPM [11] *should* be trusted but not the entire platform. Recently, Akram et al. [1] have adopted the term in a position paper on digital trust as their *vision* for trusted cloud computing using the architecture in Fig. 1.

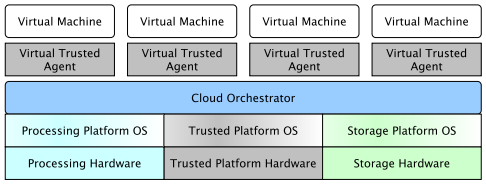

Figure 1: Proposed architecture for Trusted Computing for Cloud Computing. [1]

While the concept of processing and storage hardware and a respective management (OS) is uncontroversial and not of general concern, the notion of Trusted Platform Hardware and a Trusted Platform OS raises questions, especially, in combination with Virtual Trusted Agents (represented by vTPMs). Returning to the mathematicians' problem, it seems acceptable, even reasonable, that the Trusted Platform Hardware is in fact trustworthy and can be trusted. Specifications like [11] describe which functions and properties are required to synthesize a TPM. By inspecting a key credential, unique to each TPM, called Endorsement Key and a manufacture certificate, one can infer authenticity — assuming the manufacturer is *competent*.

2.1 Levels of Trust and Specification

The supposedly trusted platform OS seems to be a *déjà vu* for our mathematician[9]: there are no indications about the trustworthiness other than that it carries the adjective *trusted* from an external perspective. Described as quality of results, the mathematician upon interacting with the OS will need supporting evidence, or metadata other than the result itself to be convinced in every case that the interaction is correct or at least as desired or expected. This problem, although derived from an abstract case is quite intuitive: Unlike with the TPM, assuming trustworthiness of the OS is hardly justifiable. The reasons as to why that assumption is hard to substantiate are complex, especially since the specification is concerned with security. TCG's specification for Virtualized Trusted Platforms is considered a specification for *TCG Applications*. It describes general requirements such as minimum required key sizes and lengths, it has a glossary of terms, and most importantly it describes interactions with virtualization layers, or hypervisors, such as the *self-defending* Trusted Computing Base along with processes for instantiating, storing, or migrating virtual machines along with their *Virtual Trust Agents*. The idea is that a *vendor* can implement these requirements and refer to the conceived OS or platform as *Trusted* Virtualization Platform. The problem here becomes apparent however, when the predicate *trusted* derived from this specification is compared with the TPM itself: The Trusted OS along with Virtual Trusted Agents must be *fully* specified and (remotely) verifiable[13] to account for issues related to trusting software[14, 25].

2.2 Remote Attestation and Virtual Machines

From an external perspective it seems unreasonable for a specification to suggest to a suspicious party that it should simply trust whoever claims to implement the specification. While this might seem reasonable from a perspective of contracts in business to business scenarios, the nature of IoT and the idea of plug and play invalidates such *out-of-band* assurances. In order to be meaningful to a consumer or any interacting party, such a specification must provide a method to collect metadata about the potential trustee as to why it can be trusted within the scope of a specification. Defining a remote attestation process seems like a high priority task for any specification that builds on top of a TPM while requiring compliance of the implementer. Especially, since this compliance is easy to break in any software system if the potential trustee has malicious intent but also when the trustee is unaware of the fact that part of its software configuration are not suitable to an external party. The external party on the other hand would certainly like to see proof that its potential trustee implements the specification and has no additional or undesirable configuration on top of it. A focus on making a specification *attestable* yields mutual benefits both for trustor and trustee.

3 TOWARDS A USER-CENTERED ATTESTATION

Instead of defining yet another Trusted Virtualized Platform, the following subsections will explore what sort of properties and operations can lead to the conclusion that a particular platform can be *trusted*.

3.1 Principles of Remote Attestation

The need for remote attestation and its process have been defined by the TCG [11, 24]. However, besides the basic interaction with the TPM, which can be summarized as a trustworthy mechanism, an attestation should follow general principles outlined by[8]:

Principle 1. Fresh information. Information about the target of an attestation should reflect the running system, e.g. programs it is currently running and not just disk images. The reasoning is that showing only disk images tells very little about which part of this configuration is actually running. Inspired by measurement tools that provide merely measured boot, i.e. a boot sequence that reports the disk contents in the form of a hash will not supply information as to whether or not any (protection) mechanisms available on disk are actually being executed. Other measurement tools provide *load-time* information by reporting hashes of executed binaries, while in more recent years approaches for measuring active memory have

emerged. An appraiser may, however, have its own expectations as to how *fresh* the evidence is which leads to the next principle.

Principle 2. Comprehensive information. An attestation mechanism should be capable of delivering comprehensive information. This implies that such an attestation mechanism must have access to all internal states and must be able to report these using local measurement tools. While this inevitably leads to concerns about *dangerous* disclosure, i.e. reporting vulnerable states, and ultimately raises privacy questions, it also *should* consider a problem related to the amount of reference measurements in an appraisers' database — especially if there aren't any specialized appraisers.

Principle 3. Constrained disclosure. An attestation target should be able to decide which information is sent to a particular appraiser. The appraiser should be identifiable and the target platform must be able to enforce policies defining what evidence it supplies to an appraiser. Rather than suggesting the appraiser to be identifiable, it should be suggested that the appraiser reveals the type or amount of evidence it *requires* in order to be convinced. This implies that there is a semantic for attestations which leads to a conclusion as to whether or not a target can be trusted.

Principle 4. Semantic explicitness. The semantics of an appraisers' trust decision should be explicitly defined in logic. As an example on a local scale, it should define that a trust decision is made about a particular service, or the entire target platform and how trust in a particular service is established by supplying evidence of supporting or coexisting services on the target. Furthermore, it must be defined how subsequent attestations affect the initial decision and how they can be correlated for logical inferences.

Principle 5. Trustworthy mechanism. Appraisers need to infer the trustworthiness of the mechanism that is used to deliver evidence to them. This principle, although introduced last, is intuitively critical as not fulfilling it invalidates all principles so far since the evidence might simply not be convincing as it can not be trusted.

These principles were defined as general requirements for attestation architectures utilizing a trustworthy mechanism to supply convincing evidence about a particular platform. However, w.r.t. virtualization and in an Network Functions Virtualization (NFV) scenario Lauer et al. [15] have extended these principles to address issues related to the multi-user and multi-layer architecture of virtualized environments:

Principle 6. Layer linking. When a virtual environment is attested, its underlying components or layers must also be attested. Attesting a virtual platform without inspecting its underlying layers and ultimately the physical platform supplies only very limited evidence to an appraiser making a trust decision. A quote generated by a virtual trust agent such as the vTPM must be substantiated by a TPM quote so as to indicate the trustworthiness of the virtual trust agent's quote.

Principle 7. Scalability. Since attestation, i.e. (v)TPM quotes, can occur for any virtual machine triggered by any user or appraiser, substantiating each vTPM quote with a TPM quote will inevitably lead to the TPM becoming a bottleneck in a virtualization scenario. An attestation mechanism must treat the TPM as a limited and shared resource and offer a scalable protocol between vTPM and TPM quotes.

Principles 6, 7 can be seen as a virtualization specific addition to principle 5. Following these principles also reveals the proposition of this paper: Using a trustworthy mechanism implies that all other principles apply to the trustworthy mechanism itself.

3.2 Attestation in Virtualized Environments

Following these principles, two distinct approaches (or variants) have emerged in research and current standardization work. The first approach being a direct translation of remote attestation protocols using the vTPM associated with a VM as the key component of the trustworthy mechanism in combination with an attestation of lower layers using the same protocol but this time with the TPM. The later approach [15] was introduced subsequently as a solution for NFV and *X-as-a-Service* infrastructure. It assumes multiple VMs and *few* or no VM users and a single hypervisor operator. The trustworthy mechanism relies solely on the TPM while vTPMs are utilized as potentially untrusted sinks for upper layer measurements. The following paragraphs will detail these two approaches based on the principles 1-7.

3.2.1 Attestation Approach I. Attestation approach I attests virtual and physical environments using the *same* attestation mechanism (Fig. 2). An approach treating VM and lower layers equally has intuitive benefits: it is easy to integrate in an existing protocol landscape [6, 11], architectures implement vTPMs as virtual devices[3, 15], appraisers can reach a trust decision using a *standard* evaluation of the evidence or include the properties of layered systems dynamically.

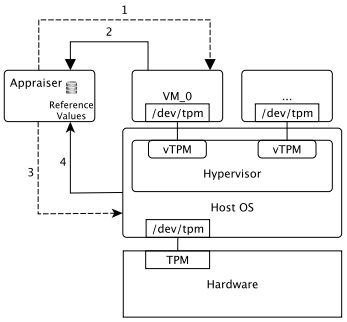

Figure 2: Attestation Approach I illustrated. The appraiser, holding *suitable* reference values, requests evidence from different layers sequentially. Boxes denote hardware modules, rounded containers denote attestable software components. Dashed lines indicate requests for meta data and bold lines indicate evidence responses.

Discussion. Separate attestation approaches are favorable for an implementer as they require little modification to existing trust establishment processes. However, this also implies specific assumptions towards the entire attestation system. Principles 1, 2 are fulfilled using what is the *de-facto* standard in Trusted Computing approaches for accessing and measuring components for later evaluation by an appraiser. In a virtualized scenario, it should be

Table 1: Fulfillment of attestation principles in a system using separate VM - hypervisor attestations.

Principle	Fulfillment
1	Requires measured or trusted boot, can accommodate Integrity Measurement Architecture (IMA) [21] for load time measurements of binaries.
2	During boot each component measures its successor into a PCR. As soon as the kernel is measured and loaded, the kernel load mechanism is responsible for measuring newly loaded binaries.
3	Appraisers do not have to be known to the target, mutual attestation is not part of any proposal.
4	The trust decision is based on whether or not supplied evidence as hash values of binaries can be found in a reference database. The appraiser then correlates attestations of any layer n with layer $n-1$. An appraisal of a VM depends on the appraisal of its hypervisor.
5	The trustworthy mechanism includes a TPM quote over a PCR (along with IMA log for hash verification) for the hypervisor. The attestation process is completed through a vTPM quote over vTPM PCRs and respective logs explaining the platform configuration hash. Credentials in a vTPM must be *trustworthy* and protected from leakage.
6	The appraiser must have *a priori* information about the locality of a VM or a list of VMs hosted by an hypervisor, a VM - hypervisor mapping. Assuming integrity and availability for such a mapping, the appraiser can correlate TPM and vTPM quotes when evaluating evidence in a decision process.
7	1:1 relation between TPM and vTPM quotes assuming each vTPM quote must be preceded or succeeded by a TPM quote (depending on principle 4,5).

noted that it is the hypervisors responsibility to maintain necessary components such as the vTPM as a root of trust for storage while the VM configuration must supply boot code that acts as the root of trust for measurement. Principle 4 and 5 are closely related and demand a rigorous definition of trust, especially, concerning the correlation of the two kinds of measurements. Explicit semantics of a trust decision are critical for a protocol design and evaluation and the traditional approach of matching well-known hashed to measured hashes is not suitable for comprehensive decisions. As far as the trustworthy mechanism is concerned, the presented approach relies on a critical assumption: VMs and associated vTPMs are not only strongly coupled but also isolated. Since the vTPM is required to perform *quotes* over its Platform Configuration Registers (PCRs), credentials must only be accessible to authorized parties such as the VM itself. However, even under the assumption of *strong* isolation, the underlying hypervisor still has responsibilities which, in an effort to provide comprehensive information, must also be reviewed. Consequently, a VM attestation must be followed by an attestation of the hypervisor — or vice versa? Intuitively, causality dictates that the hypervisor provides the run-time for a VM and therefore any property of the hypervisor affects the trustworthiness of the VM: attesting the hypervisor and subsequently attesting the VM seems effective. However, another interpretation would be that the appraisal of a VM becomes *effective* only after the hypervisor has been attested. Both interpretations seem to fulfill the attestation requirement in prose which again emphasizes the need for clear semantics.

3.2.2 Attestation Approach II. The second approach towards attestation of virtual machines is distinctly different from separate, legacy attestations insofar as it is intended for virtualized infrastructure such as *X-as-a-service* and NFV. Entitled "Hypervisor-based Attestation", the approach relies solely on the TPM and mechanisms recorded into PCRs to collect and supply evidence to an appraiser — this implies that even the *trustworthy mechanism* can be verified. The vTPM itself does not need to perform a quote over internal data structures and therefore, with attestation in mind, does not need to be raised to a *somewhat* trusted level. The attestation flow is outlined in Fig. 3.

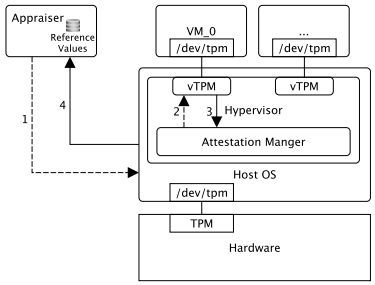

Figure 3: Hypervisor-based Attestation. The Appraiser attests the Target including *n*-VMs. Attestation Manager, a process in the Host OS or hypervisor, collects individual evidence produced by VMs from a vTPM interface. Subsequently, the attestation target sends a response containing its own and each VMs evidence.

Discussion. Hypervisor-based attestation approaches (Fig. 3) have three common benefits: Bottom-up attestation, coupled hypervisor-VM attestations, and inherently *en bloc* evaluation. Regarding the trustworthy mechanism (Principle 5), a hypervisor-based, bottom-up attestation is, at a glance, uncontroversial and in-line with conventional attestation strategies. However, common approaches use IMA [21], a kernel-based loader extension, exclusively once the module is loaded to measure and propagate loaded components to the TPM. An attestation manager or vTPM will, instead of reporting to the TPM, keep *own* records - the integrity of this functionality is then appraised using the load time hash of these components.

Table 2: Fulfillment of attestation principles in a system using hypervisor-based attestations.

Principle	Fulfillment
1	Same as Table 1.
2	Same as Table 1.
3	Same as Table 1.
4	Supplied hash-logs are compared against *golden values*, unknown values indicate untrustworthy states.
5	The trustworthy mechanism is comprised of a two stage measurement process. The first stage measures hypervisor components, including the attestation manager and vTPMs. The second stage measures VM components into vTPMs. For reporting, only the TPM is consulted and vTPM PCRs are collected and attached to a hypervisor attestation.
6	Once the appraiser has verified the attestation manager and vTPMs, the inspection of VM measurements reveals the hypervisor - VM mapping. Prior knowledge of VMs and hypervisor mappings is not required.
7	1:n relation between TPM and VM attestations.

As a result, the trust that can be placed in the completeness and correctness of such records is transitive and relies on trusting supporting mechanisms such as the attestation manager and vTPM instance. In fulfillment of Principle 4, such considerations should be addressed by a defined trust decision process that includes and respects the concept of *transitive* trust relationships. Furthermore, [15] does not specify how individual VM appraisals affect trust decisions for other VMs running on the same platform — a lack carried over from utilized measurement architectures.

3.3 Observations

Having revisited both attestation approaches, the following observations can be made: (i) Attestation of a layer inadvertently requires attestation of the layers below. (ii) Even if the Virtual Trusted Agent, or vTPM, is used for a signature, the trustworthiness of that signature relies on the appraisal of the hosting layer. (iii) Neither approach is semantically explicit with respect to their treatment of evidence in layered systems beyond suggesting exhaustive platform hash comparisons. (iv) Designated appraisers are present and capable of correlating and evaluating supplied evidence. (v) Most importantly, knowledge and connectivity of lower layers is assumed.

4 USER-CENTERED ATTESTATION

The observations of Section 3.3 will serve as defining considerations of the attestation strategy entitled *User-Centered Attestation* (UCAS) (Tab. 3).

Table 3: User-centered attestation approach in fulfillment with attestation principles.

Principle	Fulfillment
1	Measured Boot, IMA [21] providing load time integrity.
2	Causally ordered, bottom-up event log.
3	Peer-to-peer trust propagation, no attestation authorities.
4	Levels of trust, transitive relations, effects of partial attestations, and use of safe-guards.
5	VMs should be capable of gathering information of lower layers and supply them to peers acting as appraisers.
6	Published VM evidence must contain proof that the included lower layer evidence is related, i.e. the lower layer has created or is currently hosting the VM.
7	1:n relation between TPM quotes and upper layer attestations, TPM quotes could be published periodically such that quotes of upper layers can be related to them according to the semantics in principle 4.

Discussion. Unlike separate and hypervisor-based attestation, UCAS assumes connectivity only to a VM. Appraisers, which could be any device in an IoT scenario, must therefore receive evidence of the VM running a service in question along with information of its lower layers such that their appraisal is based on comprehensive information. Such evidence must be gathered bottom-up so as to reflect causalities like boot order and relations between components loading or instantiating other components. Additionally, changes to localities of VMs or changes in a lower layer must be measured, recorded, and reported accordingly. In order to enable attestations initiated by VMs and with VMs as the only point of contact for appraisers, lower layer information must be propagated such that the evidence a VM supplies automatically reflects a current state.

As far as the semantics are concerned, trusting a VM must be treated as a decision an appraiser has to make. Components involved in measurement and reporting processes in layers above trustworthy hardware may not carry the predicate *trusted* by default. Based on this notion, trust levels for components and layers can be assigned to facilitate a non-binary notion of trust and trustworthiness. Using a non-binary and layer specific notion of trust enables reasoning about the trustworthiness of supplied evidence. As a first step towards semantic explicitness, vTPM quotes will hold trust levels based on the dependencies of the vTPM component itself and ultimately appraisers can decide whether the trust level of an attestation is sufficient or if further evidence is required to resolve issues. Having this explicitness will in turn further aid scalability issues, both for evidence gathering and evaluation.

Finally, explicit semantics in combination with trustworthy mechanisms will serve to derive trust propagation mechanisms. Trust

propagation itself is essential to decentralized systems as individual nodes might not be directly accessible to a *curious* party. Being able to trust appraisals, effectively reusing it, significantly reduces the amount of individual attestation requests to nodes and remedies scalability issues related to TPM quotes.

5 RELATED WORK

Further concerns related to managing trust in Trusted Computing specifications are discussed in [26]. The semantics of trust, trustworthiness, and trust establishment in peer-to-peer systems are discussed further in [4, 18, 19]. Alternative approaches towards trust and trust management based on social notions and reputation systems are presented in [12, 16, 20].

6 CONCLUSION

Virtualization poses an interesting issue for specifications towards trustworthy systems as the *trust* placed originally only in hardware components needs to be extended to reporting and measurement mechanisms in upper layers. While approaches towards trust establishment exist, their semantics are ambiguous and an appraiser has to decide whether a virtualization platform or upper layers are trusted without much guidance or support in reasoning for such a decision. Furthermore, existing attestation approaches imply a particular topology, connectivity, and capability that does not reflect decentralized systems. A *User-Centered Attestation*, as a novel attestation system, encompasses these concerns and proposes a strategy for specifying and synthesizing suitable trust establishment mechanisms and hopefully inspires further research and contributions towards standards for open and collaborative trustworthy systems.

ACKNOWLEDGMENTS

The authors should like to thank the members of TCG's Virtualized Platform Work Group for their support and help in gaining insights in recent developments and future directions as well as the anonymous reviewers of this paper.

REFERENCES

[1] R. N. Akram and R. K. L. Ko, "Digital trust - trusted computing and beyond: A position paper," in *2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, Sept 2014, pp. 884–892.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, Apr. 2010. [Online]. Available: http://doi.acm.org/10.1145/1721654.1721672

[3] S. Berger, R. Cáceres, K. A. Goldman, R. Perez, R. Sailer, and L. van Doorn, "vtpm: Virtualizing the trusted platform module," in *Proceedings of the 15th Conference on USENIX Security Symposium - Volume 15*, ser. USENIX-SS'06. Berkeley, CA, USA: USENIX Association, 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1267336.1267357

[4] T. Beth, M. Borcherding, and B. Klein, "Valuation of trust in open networks," in *Proceedings of the Third European Symposium on Research in Computer Security*, ser. ESORICS '94. London, UK, UK: Springer-Verlag, 1994, pp. 3–18. [Online]. Available: http://dl.acm.org/citation.cfm?id=646645.699041

[5] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, ser. MCC '12. New York, NY, USA: ACM, 2012, pp. 13–16. [Online]. Available: http://doi.acm.org/10.1145/2342509.2342513

[6] E. Brickell, J. Camenisch, and L. Chen, "Direct anonymous attestation," in *Proceedings of the 11th ACM Conference on Computer and Communications Security*, ser. CCS '04. New York, NY, USA: ACM, 2004, pp. 132–145. [Online]. Available: http://doi.acm.org/10.1145/1030083.1030103

[7] M. Chiang and T. Zhang, "Fog and iot: An overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, Dec 2016.

[8] G. Coker, J. Guttman, P. Loscocco, A. Herzog, J. Millen, B. Hanlon, J. Ramsdell, A. Segall, J. Sheehy, and B. Sniffen, "Principles of remote attestation," *Int. J. Inf. Secur.*, vol. 10, no. 2, pp. 63–81, Jun. 2011. [Online]. Available: http://dx.doi.org/10.1007/s10207-011-0124-7

[9] E. W. Dijkstra, "Programming considered as a human activity," in *Classics in Software Engineering*, E. N. Yourdon, Ed. Upper Saddle River, NJ, USA: Yourdon Press, 1979, pp. 1–9. [Online]. Available: http://dl.acm.org/citation.cfm?id=1241515.1241516

[10] P. Garcia Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino, A. Iamnitchi, M. Barcellos, P. Felber, and E. Riviere, "Edge-centric computing: Vision and challenges," *SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 5, pp. 37–42, Sep. 2015. [Online]. Available: http://doi.acm.org/10.1145/2831347.2831354

[11] ISO, "Trusted platform module library," International Organization for Standardization, Geneva, Switzerland, ISO ISO/IEC 11889-1:2015, 2015.

[12] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in *Proceedings of the 12th International Conference on World Wide Web*, ser. WWW '03. New York, NY, USA: ACM, 2003, pp. 640–651. [Online]. Available: http://doi.acm.org/10.1145/775152.775242

[13] G. Klein, K. Elphinstone, G. Heiser, J. Andronick, D. Cock, P. Derrin, D. Elkaduwe, K. Engelhardt, R. Kolanski, M. Norrish, T. Sewell, H. Tuch, and S. Winwood, "sel4: Formal verification of an os kernel," in *Proceedings of the ACM SIGOPS 22Nd Symposium on Operating Systems Principles*, ser. SOSP '09. New York, NY, USA: ACM, 2009, pp. 207–220. [Online]. Available: http://doi.acm.org/10.1145/1629575.1629596

[14] B. W. Lampson, "A note on the confinement problem," *Commun. ACM*, vol. 16, no. 10, pp. 613–615, Oct. 1973. [Online]. Available: http://doi.acm.org/10.1145/362375.362389

[15] H. Lauer and N. Kuntze, "Hypervisor-based attestation of virtual environments," in *Advanced and Trusted Computing (ATC), 2016 Intl IEEE Conferences*. IEEE, 2016, pp. 333–340.

[16] S. Marti and H. Garcia-Molina, "Taxonomy of trust: Categorizing p2p reputation systems," *Comput. Netw.*, vol. 50, no. 4, pp. 472–484, Mar. 2006. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2005.07.011

[17] R. Morabito, "Virtualization on internet of things edge devices with container technologies: A performance evaluation," *IEEE Access*, vol. 5, pp. 8835–8850, 2017.

[18] M. Richardson, R. Agrawal, and P. Domingos, "Trust management for the semantic web," in *Proceedings of the Second International Conference on Semantic Web Conference*, ser. LNCS-ISWC'03. Berlin, Heidelberg: Springer-Verlag, 2003, pp. 351–368. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-39718-2_23

[19] P. D. Rowe, *Bundling Evidence for Layered Attestation*. Cham: Springer International Publishing, 2016, pp. 119–139. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-45572-3_7

[20] A. Ruan and A. Martin, "Repcloud: Achieving fine-grained cloud tcb attestation with reputation systems," in *Proceedings of the Sixth ACM Workshop on Scalable Trusted Computing*, ser. STC '11. New York, NY, USA: ACM, 2011, pp. 3–14. [Online]. Available: http://doi.acm.org/10.1145/2046582.2046586

[21] R. Sailer, X. Zhang, T. Jaeger, and L. Van Doorn, "Design and implementation of a tcg-based integrity measurement architecture." in *USENIX Security Symposium*, vol. 13, 2004, pp. 223–238.

[22] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, Oct 2016.

[23] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, 2016.

[24] TCG, "Virtualized trusted platform architecture specification," Trusted Computing Group, Beavertion, OR, USA, TCG v1r26:2011, 2011.

[25] K. Thompson, "Reflections on trusting trust," *Commun. ACM*, vol. 27, no. 8, pp. 761–763, Aug. 1984. [Online]. Available: http://doi.acm.org/10.1145/358198.358210

[26] *Trusted Multi-Tenant Infrastructure Reference Framework*, Trusted Computing Group, 12 2013, rev. 1.

[27] S. Yi, C. Li, and Q. Li, "A survey of fog computing: Concepts, applications and issues," in *Proceedings of the 2015 Workshop on Mobile Big Data*, ser. Mobidata '15. New York, NY, USA: ACM, 2015, pp. 37–42. [Online]. Available: http://doi.acm.org/10.1145/2757384.2757397