Privacy in Distributed Commercial Applications

Nicolai Kuntze, Carsten Rudolph

Privacy in Distributed Commercial Applications[1]

Nicolai Kuntze and Carsten Rudolph

Fraunhofer Institute for Secure Information Technology SIT,
Rheinstrasse 75, 64295 Darmstadt, Germany
nicolai.kuntze@sit.fraunhofer.de, carsten.rudolph@sit.fraunhofer.de

Abstract. Devices installed in end-user's homes but controlled by network operators can be the basis for advanced distributed commercial applications. Virtual machines on these devices can be used to efficiently deploy and manage such applications provided by various competing entities. This paper discusses some security and privacy requirements of such distributed commercial applications and proposes two different approaches to root security and privacy in hardware-based attestation of nodes and virtual machines.

Keywords: Virtualisation, Privacy, Trusted Computing, Attestation.

1 Introduction

A major part of commercial applications on the Internet are currently client-server based with browser software representing the application on the client side. Individual users are identified via user name and password with additional security measures in critical applications such as electronic banking. Privacy requirements are often violated through the use of cookies, reading of browsing history, or linking of data between different services or providers. We currently see a diversity of alternative approaches to commercial applications appearing on the market. Some of these approaches involve third-party applications being installed on the client side, i.e. on the computer of the end-user. One prominent example is the App Store for iPhone applications [2] where applications are installed on the device. These applications can enable additional services. Also, devices installed by network operators can pave the way for new approaches to distributed commercial applications, in particular for the distribution of multi-media content.

An architecture for distributed commercial applications is being developed in the EU funded Nanodatacenters Project (NaDa). Autonomous devices are established in the households of the end users. These devices belong to a network operator (NO) and remain under his administrative control. Content providers (CP) can pay for resources on the devices in order to run applications to distribute their actual content according to their needs in terms of security and distribution strategies. P2P protocols are used

[1] This work was partly accomplished within the project Nanodatacenters under the grant number 223850 financed by the EC.

to distribute the content in the network operated by the NO allowing for an efficient utilization of the network and its bandwidth.

In such a scenario it is not sufficient to protect privacy on the level of bi-lateral relations such as those between client and server. Privacy concerns arise if one identity is used in different applications, or different stake-holders and applications share the same devices and networks. Collecting and relating data or linking activities of a particular entity become possible. Aggregated knowledge may bring NO or CP in a position where they could predict an end-user's behavior, impersonate the user, or learn other confidential information. Such privacy issues are also relevant for the distribution system developed in NaDa.

This paper discusses the different and possibly conflicting security and privacy requirements of the stakeholders in distributed commercial applications. Pseudonymous or anonymous attestation of the state of a device protected by a hardware security anchor as part of Trusted Computing technology can be one building block towards privacy-preserving secure distributed commercial applications. Using the example of the NaDa nodes and P2P network, two approaches to pseudonymous and anonymous attestation are discussed. This attestation must include the state of the node itself as well as the CP application running in a virtual machine. The first approach uses a generic method of deep attestation through a virtual machine to the hardware TPM. The second proposal is tailored for the NaDa P2P network and relies on the mutual attestation of nodes in the network. Finally, this paper briefly discusses the two different approaches of attestation to different stakeholders in respect of their particular privacy and security requirements.

2 NanoDatacenters Architecture

The Nanodatacenters project aims at the virtualization of content distribution and access infrastructure. Inexpensive nodes form a P2P overlay network. On each node the network operator NO offers content providers CP virtual machines for rent, so-called slices. A CP can use such a slice to install operating system and application software. The underlying node uses hypervisor architecture to control slices and to provide interfaces to the slices and to the end-user. All communications and other external interfaces are only available through the hypervisor and thus controlled by the NO. One main idea of the NaDa architecture is to provide a network of nodes that can be used by the CP applications to efficiently manage a large number of client applications. The focus is on the domain of multi-media distribution. In this use-case slices can use resources on the node to store and cache multi-media data so that requested data might be provided directly through peers in the network. This approach can minimize the required bandwidth for a central server. Functional prototypes for such a P2P network are being developed in the NaDa project.

In this paper we assume that the operating system on the node, as well as the hypervisor and the operating system and application software on the slice, is properly implemented and free of failures as long as there are no malicious changes to software or configuration. Note that in order to increase assurance in the node's software the

NO may require validation and certification on all software and applications running on the nodes.

Security requirements of the stakeholders user, NO and different CPs, are discussed in the next section.

3 Security Requirements Analysis

The different stakeholders, namely user, NO and CPs have possibly conflicting security and privacy requirements in the scenario described above. The presented application area bears similarities to concepts for outsourcing resources for services; for example, in cloud computing scenarios. Different parties move their computational, storage and communication load to an operator who then provides the resources and returns the results to the customer or directly to the end-user via clearly specified user interfaces. Advanced examples are operated e.g. by Amazon where customers can also run publicly available web servers on distributed leased resources. However, in contrast to these applications the P2P character of the underlying network and of possible applications results in unclear boundaries. Furthermore, locating all nodes at the homes of the end-user means that the NO has no physical control over the nodes. The following paragraphs discuss the different security requirements for user, NO and CP.

3.1 User Requirements

The end user represents a special case in this analysis due to his interaction with the node by receiving content and paying for it through the node. Privacy relevant information for the user includes data on content usage but also usage of the Internet through other services not located on the node. Content usage includes the content provided by a particular CP but also from the other CPs on his device. Also, end user specific data like payment information may be stored on the node associated with the user and the physical location of the particular box identifying the address of the user's household. All this information could support the building of a personal profile on the user, thus violating the user's privacy. Therefore, it is a very strong requirement to provide isolation mechanisms preventing unpermitted profiling through single or co-operating content providers.

3.2 CP Requirements

In all such scenarios the isolation of all different stakeholders running their systems on a shared platform is required in terms of resources and information governance. Resource protection has to be established, preventing customers from influencing the operation of other customers sharing the same resources. Information governance states that one party should not be able to access the information stored or processed by other parties. As the NO is the owner of the platform, and therefore has full control over it, all other parties need to lay trust in him. Although some privacy properties can

also be enforced towards the NO, this needs access to data required in performing its processing tasks.

In the P2P network regarded here, nodes are operated in the households of end users and are therefore not under the direct physical control of the NO. Thus, nodes can potentially be manipulated. Establishing trust relations between the nodes involved requires that each node is able to provide proof on its state. This state shall represent the operational state as defined by the NO. As the NO is assumed to have full control over the node, it is in principle able to modify each node and with it also the applications running on it on behalf of the NO's customers. Thus, customers also need to be able to verify the state of their respective application. This proof has to include evidence on the state of their particular slices on the node as well as the state of the underlying node system.

Furthermore, content providers should not have access to information on the status of other competitors running their applications on the same node. This property must also be satisfied if content providers operate their own monitoring software as part of their application. This requirement is to prevent leakage of business relevant information protecting the revenue model of each content provider (i.e. of the customers of the NO).

3.3 NO Requirements

The proof of the state of the node is also relevant for the NO who wants to ensure that only authorized and non-manipulated nodes can access the P2P network. This also requires that the node has got some kind of identity used to sign attestation data. This identity can also identify a node towards the CP's applications in order to provide evidence that data exchanged in the network stems from proper nodes actually belonging to the NO network. However, these identities also allow the CP to identify each individual node, and therefore privacy relevant information can be directly contained in them or be related and linked. Parameters like location, configuration, amount of data, bandwidth, accounting information and utilization allow the CPs to analyze the market penetration of a particular NO. Therefore the NO requires that information on the state of the nodes given to the content providers shall be restricted, and in particular, not allow the analysis of the overall network.

In the NaDa project a security architecture is built on Trusted Computing technology. One central challenge of the architecture is the realization of attestation for node and slices on the node in regard of the different security and privacy requirements described above. The contribution of this paper is to propose two ways to achieve this attestation. The following section gives a brief overview of the underlying concepts of Trusted Computing.

4 Trusted Computing

As discussed in section 3, privacy of the node's identity and secure reporting of the system state are important security requirements. In many approaches these requirements can be contradictory. Trusted Computing TC [17] offers a hardware root

of trust that provides certain functionalities designed to approach the combination of requirements (see also [6, 8]). In this section these functionalities are introduced.

TC as defined by the Trusted Computing Group (TCG), are computer systems extended by additional components that shall bring trust to the computing environment. Trust means that components of the system always work as implemented. To achieve this goal, the TCG has published and is still working on specifications describing architectures affecting system components at any level from hardware to the operating system.

Most important hereby is the Trusted Platform Module (TPM). This module is mostly realized as a hardware chip hard-wired to the computer platform. It implements basic cryptographic functionality like SHA-1, message digest creation, random number generation, creation of 2048 bit RSA key pairs, and a RSA engine for encryption and signing purposes. Realized as an independent hardware module it provides protected capabilities for secure storage. The TCG defines three different roots of trust on which the trust to the whole system is built.

First, the Core Root of Trust for Measurement (CRTM) [9] performs initial measurements (e.g. hash values of the respective firmware) of system components involved in the booting process. Measured (and individually trusted) components can then perform measurements of other components involved in the next stage of booting. Through this principle of transitive trust, trust in the correctness of the measurement values can be passed on to the operating system, hypervisor and into virtual machines. Through this architecture it shall be guaranteed that a computer system always starts in an authenticated state that can be verified by an external entity and used for establishment of trust. This reporting process is called Remote Attestation (RA). Second, the Root of Trust for Reporting (RTR) is used in the RA process. Remote entities need to be able to validate that measurement results are genuine and come from a valid TPM and platform. For this purpose every TPM contains a unique 2048 bit RSA key pair, the Endorsement Key (EK), which is generated before shipping. The EK, together with an EK Credential, represents the identity of the platform. However, the EK is never directly used for signing attestation data. Instead, Trusted Computing provides two different techniques to conceal the identity of the platform embodied in the EK during the attestation of the platform:

4.1 Pseudonymous Attestation Based on a Privacy CA

The first proposed concept to protect the privacy of the customer introduced by TCG relies on a trusted third party issuing pseudonyms for the individual TPM platforms, the so-called privacy certification authority (Privacy CA). When a TPM needs to authenticate to a verifier, it generates an RSA key pair called the Attestation Identity Key (AIK) and sends the AIK public key together with public EK and EK certificate to the Privacy CA. There the AIK is authenticated using the EK; the Privacy CA then issues (based on certain rules) a certificate on the TPM's AIK. The TPM then uses this certificate (and the corresponding AIK) to sign measurement values. Note that in this approach verifiers cannot link different actions from the same platform as long as different AIKs are used. However, the Privacy CA can link all AIKs to the unique EK and therefore could reveal the identity of the platform.

4.2 Direct Anonymous Attestation

Direct Anonymous Attestation (DAA) as explained in [1] is a practical and efficient scheme for authenticated pseudonymous attestation with strong privacy properties. In the scheme four parties are involved as shown in Figure 1. A certification authority CA certifies long term public keys (1). The role of the CA corresponds to the role of CAs in usual public key infrastructures. The so-called DAA-issuer creates blindly signed credentials in the DAA join phase (2) that is later used in the DAA attestation (3). The main idea is that the certificate is not actually shown to the verifier. Instead, a zero-knowledge proof is used to prove that the platform owns a certificate issued by the DAA issuer. Thus, even the DAA issuer in principle cannot link different AIKs to the EK of the platform. It is to be noted that DAA allows for different operation modes with different levels of anonymity. Rudolph presents in [11] an attack showing the limitations of this scheme. This attack is also discussed in [7].

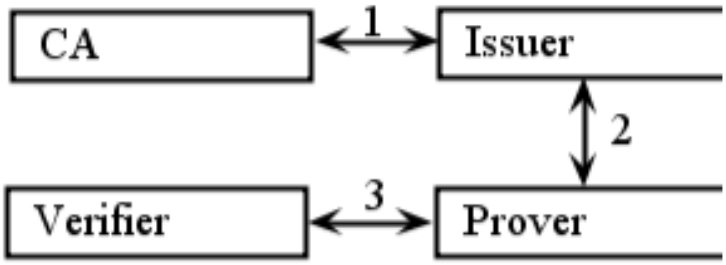

Fig. 1. Overview on DAA interactions.

5 Attestation of Virtualized Environments

Satisfaction of the security and privacy requirements identified in Section 3 depends on the state and configuration of the node, in particular of the hypervisor controlling the slices and providing the desired isolation between the slices. The design of the node operating system that is attested to the communication partners is not within the focus of this paper. Different approaches exist for the communication within distributed environments, e.g. building IP over P2P [4] communication allowing for a Virtual Private Network for each customer. There is also work on the design of virtual machine sand-boxes in overlay networks [20, 18]. For this paper we assume the existence of a secure and hardened system with known reference measurement values.

However, the existence of a secure node software is not sufficient. Different stakeholders cannot blindly trust the nodes to be properly implemented and configured. They need assurance that nodes are not changed or compromised. As described in the previous section, TC can be used for attesting the state of a node to a

remote party. Nevertheless, deploying remote attestation is not straightforward in a multi-stakeholder environment with multiple virtual machines installed in the slices of each node. Virtual machines only have limited knowledge about the state of the node they are running on. Furthermore, a single hardware TPM with a single identity has to be used for the attestation of several virtual machines. As the virtualized system has no direct access to the TPM of the underlying platform, it has to use its own (intermediate) root for security, a so-called virtualized TPM (vTPM). This concept is also introduced and discussed from different perspectives in [12, 14, 19]. The correct functionality of a TPM depends on its implementation located in the underlying virtualization system. In order to fulfill the requirements of providing proof of the state of the system, this proof also has to include proof on the underlying system.

In the NaDa architecture attestation needs to consider the individual requirements of NO, CP and end-user. The end-user has to trust the NO with respect to the correctness and security of the software installed on the node and of the configuration. Therefore, in this approach we assume that the end-user also delegates the checking of the integrity to the P2P network controlled by the NO.

The NO is mainly interested in the integrity of the node itself. The software installed within the slices is not relevant for the NO as the slices are controlled by the hypervisor. Consequently, the standard TC protocols for remote attestation can be used to check the integrity of the node each time it joins the P2P network for a new session. This attestation needs to be properly integrated into the establishment of new session keys with the peer nodes. For increased assurance, bi-lateral attestation can be repeated during a running session. Failure would result in termination of the session.

In contrast to the NO, CPs are interested in the integrity of their own slices and also that the underlying node and environment is in accordance with the SLA. Information on other slices belonging to different CPs shall not be available to a CP. Furthermore, as described in Section 3 the CP should not get all possible status information because some of the information can be considered part of the confidential business information of the NO. It might also be required that different CPs shall not be able to link status information from their different slices to platform identities.

The following subsections present two different approaches for attesting node and virtual machines in the slices. The nodes in the NaDa project only allow one layer of virtualization. Thus, the more complex case of several virtualization layers is not discussed here. Considering different concurrent attestation requests on one node can result in a performance bottleneck. Fuchs et al. analyzed in [15] possible strategies to approach this problem.

In addition to the attestation it is necessary to protect the communication channels between nodes as well as between slices. These channels need to be bound to the hardware security anchors such that it is assured that the attested state is indeed the state of the end-point of the communication channel. Goldman et al. first proposed a protocol that binds the secure channel established by SSL/TLS to the remotely attested platform [5]. Other work in this area also shows the importance of this ability e.g. [3, 10, 13].

5.1 Pseudonymous Node Using Deep Attestation

In this approach the status of one slice as well as the status of the underlying platform are attested in a single step towards another slice. This other slice is usually located on a different platform, but the slices can also share one platform. The signature generated by the TPM as result of the TPM_Quote command is included in the quote produced by the vTPM.

During the deployment process of a new slice on the node two individual attestation identity keys are created that have to be assumed as being unique for this slice. The first AIK is located in the TPM of the node, while the second key (called vAIK) is an attestation identity key of the vTPM of this particular slice. The usage of the vAIK is restricted by the TPM_Bind operation controlled by the TPM to a specific state of the node and can be bound through the vTPM to a particular state of the individual slice. A certificate for the AIK is issued by the NO or a representative stating that the node is a node of a certain type and that its usage is restricted to a particular role in the P2P network. The certificate for the vAIK needs to include the AIK of the node to bind the vAIK and the vTPM to the physical TPM.

The slice to be attested has to provide proof that the system running within the slice is compliant to its desired state according to the implementation deployed, and that the node hosting the slice is also not altered. Integrity of the node is in particular needed to show that the vTPM is operating according to its definition and to ensure the isolation of slices on the node.

By using a unique AIK for each slice on a node it is ensured that the identity of the node cannot be linked to other attestation messages for other slices on the same node. The node guarantees that the AIK usage is limited to the attestation of one particular slice.

The attestation process is denoted with deep attestation because it goes through the slice and the vTPM to the hardware TPM of the node. Thus, in one step, it provides two encapsulated remote attestation protocols of the type. During an attestation process a verifier sends to the slice a remote attestation challenge that results in a vTPM_Quote command to the vTPM. The challenge contains a nonce generated by the challenger and used to prevent replay attacks. Implementing deep attestation uses this nonce to also challenge the TPM of the hosting system using TPM_Quote and to return in the result of the vTPM also the attestation result from the TPM.

In order to gain better scalability, the PCA required to sign the AIKs for each slice could be implemented as an internal service of the node operating as a representative of the NO. The required credentials to sign the AIKs are shared between all nodes or a certain subset of nodes. Thus, credential information cannot be used to track back a certain AIK signature to a specific node. To ensure that the node is not issuing fake certificates, and to prevent extraction of the credentials, certain means are required. First, the credentials have to be bound to a certain state of the node preventing malicious software on the node to make use of them. Also, the lifetime of these credentials needs to be limited to a certain (relatively short) time. After this time a replacement mechanism in the P2P network has to renew these credentials and new certificates for the AIKs should be issued.

Clearly, a so-called privacy certification authority (PCA) with high availability is required to issue new AIK certificates for each slice quite often. There are strong

security requirements for this PCA, as it is able to reveal the public EK of the TPM for each AIK. Thus, the PCA is able to link different AIKs of the same platform. High availability and security are often seen as contradictory (or at least expensive) requirements. Furthermore, the PCA needs to be separated from the stakeholders of the system. Therefore, it is not clear that a PCA could run with a reasonable business model in such a system. Using DAA could overcome this problem and also provide better privacy.
The deep attestation approach provides attestation of one slice and the underlying node in a single step. The next proposal distinguishes between node attestation and slice attestation. It is tailored for the use in the NaDa P2P scenario.

5.2 Two Step Attestation

The vTPM, in order to implement the deep attestation, needs to differ from a standard TPM as it has to include a TPM_Quote result from the hardware TPM in its own quote. Further, the hardware TPM needs to provide a unique AIK for each slice. Splitting the process in two steps leads to a scheme that does not require modifications to the TPM and also requires only one identity per node and one identity per slice.

Figure 2 depicts the high level steps required. Step one is performed between the hosting environments of the nodes in the P2P network. This process establishes the fundamental trust and also allows that all services offered by the host to the slices are trustworthy as soon as the slice is part of the network. This can be done initially for each session and includes a key exchange to build a session as it is e.g. proposed in [16]. It should be notes that each CP will control at least one node of the peer-to-peer network in order to be able to introduce data into the network. For the CP this node is the root for establishing trust in the other nodes of the network. This trust is achieved by mutual attestation between nodes in the network together with trust in the node software and configuration. The only way to communicate for the slices is through this (now trusted) overlay network. Thus, communication with another slice implies that the underlying host platform is part of this P2P network.
In step two the CP checks the integrity of the slices by performing a remote attestation using vTPM on the slice. These vTPM can operate according to the defined TPM standard.

The host can restrict the interaction of slices according to interaction policies, allowing only for communication between slices from the same owner. This forms another logical overlay network. Interaction between slices is isolated between different CPs. Thus, privacy breaches on the level of information flows between slices are prevented.

In comparison with deep attestation several advantages and disadvantages can be identified. First, it can be seen as a disadvantage that the content providers must trust the NO to enforce the attestation-based control for nodes joining the P2P network. However, the NO will also provide reference values for node states to be used in deep attestation. Therefore, if CPs can actively check the state of the nodes in the case of deep attestation they still have to trust the NO in providing secure configurations.

The option of having private logical overlay networks can be used to isolate traffic and increase the privacy of the CPs. Furthermore, it is not necessary to make the actual identity of nodes visible to the CP. For many applications it is sufficient to know that the slice is running on a node in the P2P network.

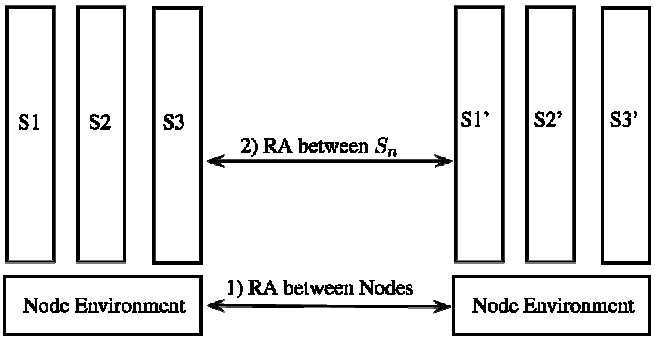

Fig. 2. Two step remote attestation.

This second solution integrates the attestation of the nodes into the establishment of the P2P network. Optional subsequent slice attestation is in the control of the CP and independent from the underlying node controlled by the NO. Furthermore, only one AIK is needed for each node. Thus, organizational processes are more efficient. This solution was chosen for integrating it into the NaDa architecture.

6 Conclusion

The different types of attestation for virtual machines show that Trusted Computing mechanisms can be adapted to different scenarios and requirements without changes to the underlying trusted platform architecture. Different privacy requirements can be satisfied on the basis of this technology without losing security properties required by network operators and service providers. It is this combination of privacy and security that makes trusted computing based solutions suitable for various electronic commerce applications. The use of TPMs in home boxes such as those used in the NaDa project can increase the security of these boxes, and also be the basis for reliable devices to be used in advanced distributed electronic commerce applications.

Prototypical implementations of the privacy and security mechanisms, including independent remote attestation of nodes and slices, have been completed within the Nanodatacenters project. Future work will include the integration of these mechanisms into distributed electronic commerce architectures in order to evaluate the overall privacy and security properties, and the efficiency and usability, of the architecture.

References

1. Brickell, E., Camenisch, J., Chen, L.: Direct anonymous attestation. In: Proceedings of the 11th ACM conference on Computer and communications security, pp. 132–145. ACM New York, New York (2004).
2. Chen, G., Rahman, F.: Analyzing Privacy Designs of Mobile Social Networking Applications. In: Proceedings of the 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, vol. 2, pp. 83–88. IEEE Computer Society (2008).
3. Dietrich, K.: A Secure and Reliable Platform Configuration Change Reporting Mechanism for Trusted Computing Enhanced Secure Channels. In: ICYCS 2008 – The 9th International Conference for Young Computer Scientists, pp. 2137–2142 (2008).
4. Ganguly, A., Agrawal, A., Boykin, P., Figueiredo, R.: IP over P2P: Enabling Self-configuring Virtual IP Networks for Grid Computing. In: Proc. International Parallel and Distributed Processing Symposium (2006).
5. Goldman, K., Perez, R., Sailer, R.: Linking remote attestation to secure tunnel endpoints. In: Proceedings of the first ACM workshop on Scalable trusted computing, pp. 21–24. ACM New York, Ne York (2006).
6. Kuntze, N., Schmidt, A. U.: Trusted ticket systems and applications. In: New Approaches for Security, Privacy, and Trust in Complex Systems. Proceedings of the IFIP sec2007. Sandton, South Africa 14-16 May 2007. Springer-Verlag (2007).
7. Leung, A., Chen, L., Mitchell, C.: On a Possible Privacy Flaw in Direct Anonymous Attestation (DAA). Technical report, Tech. Report RHUL-MA-2007-10, Mathematics Department, Royal Holloway, University of London, December (2007).
8. Mitchell, C. et al.: Trusted Computing, pp. 1. IEEE Press, London (2005).
9. Pearson, S.: Trusted Computing Platforms, the Next Security Solution. Bristol UK: HP Laboratories (2002).
10. Pollock, W., Pitcher, C. Chicago, I.: Identifying Trustworthy Hosts Using Remote Attestation (accessed on 31st May, 2010), http://facweb.cti.depaul.edu/ctiphd/ctirs04/submissions/camera-ready/Pollock.doc
11. Rudolph, C.: Covert Identity Information in Direct Anonymous Attestation (DAA). IFIP Security, pp. 232:443 (2007).
12. Scarlata, V., Rozas, C., Wiseman, M., Grawrock, D., Vishik, C.: TPM Virtualization: Building a General Framework. Trusted Computing: Ein Weg zu neuen It-sicherheitsarchitekturen (2007).
13. Schechter, S., Greenstadt, R., Smith, M.: Trusted computing, peer-to-peer distribution, and the economics of pirated entertainment. In: Proceedings of The Second Annual Workshop on Economics and Information Security, pp. 29–30. Springer (2003).
14. Schmidt, A., Kuntze, N., Kasper, M.: On the deployment of Mobile Trusted Modules. In: Proceedings of the 9th IEEE Conference on Wireless Communications and Networking (WCNC 2008), pp. 3169–3174.

15. Stumpf, F., Fuchs, A., Katzenbeisser, A., Eckert, C.: Improving the scalability of platform attestation. In: Proceedings of the 3rd ACM workshop on Scalable trusted computing, pp. 1–10. ACM New York, New York (2008).
16. Stumpf, F., Tafreschi, O., Roder, P., Eckert, C.: A Robust Integrity Reporting Protocol for Remote Attestation. In: Proceedings of the Workshop on Advances in Trusted Computing (WATC) (2006).
17. Trusted Computing Group. TPM Specification Version 1.2 Revision 103. Trusted Computing Group (2009).
18. Tsugawa, M., Fortes, J.: A Virtual Network (ViNe) Architecture for Grid Computing. In: Proc. of the IEEE Intl. Parallel and Distributed Processing Symp. (IPDPS), Rhodes, Greece (June 2006).
19. Winter, J.: Trusted computing building blocks for embedded linux-based arm trustzone platforms. In STC '08: Proceedings of the 3rd ACM workshop on Scalable trusted computing, pp. 21–30, ACM, New York (2008).
20. Wolinsky, D., Agrawal, A., Boykin, P., Davis, J., Ganguly, A., Paramygin, V., Sheng, Y., Figueiredo, R.: On the Design of Virtual Machine Sandboxes for Distributed Computing in Wide-area Overlays of Virtual Workstations. In: Virtualization Technology in Distributed Computing, p.8. VTDC (2006).