Nicolai Kuntze, Jürgen Repp
Fraunhofer SIT, Rheinstrasse 75, 64295 Darmstadt
{Nicolai.Kuntze|juergen.repp}@sit.fraunhofer.de
Martin May, Fabio Picconi
Thomson, Paris
{Martin.May|Fabio.Picconi}@thomson.net
Renata Teixeira
Lip6, Paris
renata.cruz.teixeira@gmail.com

TRUST IN THE P2P DISTRIBUTION OF VIRTUAL GOODS

Abstract: Distribution of virtual goods over the IP based infrastructure offered by the Internet requires efficient techniques w.r.t. to the utilization of the existing resources. One approach here applies methods from the peer to peer domain to ensure that the required traffic mostly is situated in the cost efficient last mile. This paper presents this approach and discusses the security implications.
The proposed security and trust architecture includes solutions for integrity protection of data as well as for software on the device, exclusion of manipulated nodes from the network, and isolation between applications by different stakeholders residing in parallel on the same platform. All solutions can be build on existing secure hardware anchors as provided by the Trusted Platform Module (TPM) and its certification infrastructure.

Keywords: Peer to peer, media distribution, security, trust

1. Introduction

Distribution of virtual goods in the Internet has relied on a client server model for the last years. A central entity provides the goods and the clients are only receiving the content. Different systems like the well known iTunes are based on this paradigm. Aside this traditional approach of service delivery distributed solutions arose like distributed caching, Content Distribution Networks (CDNs) [1], and more recently peer-to-peer (P2P) networks as it is for example shown in [2].

Today’s cost structure of the infrastructure operated by the Internet Service Providers (ISPs) is not optimal for the traditional client server distribution model. The bandwidth on the side of the backbone infrastructure is highly expensive as its utilization is already high. On the other side it is recognized that the part of the infrastructure physically near to the end user, sometimes referred as last mile, is not utilized at this amount. Therefore it is desirable from the point of view of the ISPs to push to traffic associated with the content distribution to a certain extend into this part of the infrastructure and also to reduce the traffic between the peers. This results in distribution networks based on nodes located at the edges of the ISP networks situated in the households of the end users.

Such a network design allows for optimized bandwidth utilization in the core network by applying optimized P2P protocols that take the physical configuration of the ISP network into account and offers proper caching and data propagation methods. Additionally costs for expensive central server systems can be reduced as the complexity of these systems is diminished due to the abilities inherent to a P2P network like redundancy and load balancing. Also of interest is an improved Quality of Experience (QoE) considering a video on demand (VoD) use case. There, the network is able to store the data expected to be viewed on the device of the end user or in his neighborhood. Such functionality is based on certain consumption data know and analyzed by the overall system operated by the ISP.

As the ISP is not the owner of the virtual goods in most of the cases his aim is to provide an interesting platform for the actual owners of the content and to allow each separate Content Provider to offer and protect their content according to their needs. It is important to see here that the goods offered may be very different in their needs as e.g. an online game has different requirements then a VoD service. Therefore different (concurrent) offerings are hosted in the same node at the same time. Sharing of one environment between different customers is already a well known business case like Web based storage services, Web Email, or resizable computing capacity as offered e.g. by Amazon.

The EU funded research project Nanodatacenters (NaDa) develops a common platform that provides a basic set of functionalities to establish a trustworthy environment at the side of the user households that can be used by the Content Providers to offer their goods to the users. One underlying decision is that the nodes belong to the respective ISP and is under the administrative control of him. On top of this node the ISP offers slices to the Content Providers where they can establish a software system that allows for their special business case and the needed level and methods of security. The end user provides only an environment, namely power and connectivity, where the node can operate in and is able to consume content offered by the node. Due to the caching and load balancing methods that are part of the research of NaDa content is dispersed to the nodes and redistributed by them. This also implies that it happens that content is stored on nodes who are not delivering content to the end users in “their” households but caching it for their neighbors. The caching strategy only optimizes the QoE but not the storage location.

NaDa defines a secure platform that provides defined interfaces and methods for security, monitoring, and communication that is used by the content providers on each node. This platform is designed to protect the content that is injected into it. At the moment content is moving out of the platform due to the business model of a Content Provider possibilities exist also to protect it e.g. by means of watermarking or digital rights management. The responsibility to decide on the appropriate protection scheme is at the side of the content owner.

This paper presents the security architecture that is currently under development in the NaDa project supporting the special requirements that arise from the P2P approach. Especially the application of P2P technology in a commercial environment with high value data bears new challenges that need to be addressed by the architecture as it is operated in a potential hostile environment. We will show how to use an already available hardware trust anchor to spur the trust establishment by providing a appropriate level of security.

This paper is organized as follows. In chapter 2 the security requirements analysis base on the use case is presented. Architecture, Use Case story and functional building blocks are presented in chapter 3. The paper closes with a short conclusion showing the next planed steps in the research process.

2. Security Requirements Analysis

Security in the given use case has the aim to establish trust of the customers as well as the ISPs into the architecture. This chapter shows the different security aspects to handle by the architecture that are derived from the business process of the use case scenario that is presented in chapter 3.2 (see also [3]). Not covered in this chapter are perils resulting from faults in the software.

Aspects of information governance like authenticity of a node acting within the overall network, integrity of this node, traceability, integrity or confidentiality of data that is transported between nodes involved, or the enforcement of certain licenses belonging to the content distributed are included in the set of security requirements for which basic security mechanisms need to be provided by the architecture. In this paper we concentrate on the central requirements, namely the security properties of isolation of the different stakeholder applications but also the ISP provider’s privacy, the identity of each node, status of the node, and the integrity of the content and introduce them in the following. Aside of these security requirements is availability also of interest and is to be considered in the design of the architecture by reducing the number of components whose failure may lead to a breakdown of the network.

One important aspect in NaDa is the resource efficiency of the overlay network in terms of bandwidth utilization as introduced above. The ISP is interested to concentrate the activities of the nodes in the network next to the node reducing the load on the backbone. To accomplish this task the P2P routing mechanisms require detailed data on the location of the nodes, latencies between the nodes, and the installed physical network. Also utilization information are required e.g. to determine the right time to distribute content and to save costs but providing the expected QuE for the end user.

All these data describe the business base of the ISP and are therefore considered highly sensitive. A concurrent ISP (maybe entering the market) could analyze using these information the high revenue system, cost structures, and isolate interesting areas to enter the market. If a party having access to these information works together with a competitor he could hand over them to them for a certain price endangering the business model of the ISP. A second issue is the incentive for the customer to respect the desired routing and caching aim. If he is only interested in serving his end customers then he will not care for optimal bandwidth usage.

Each customer operating his slices has his own interests and therefore stores data on each node with different levels of required protection. Some of the stored data a most relevant for the operation of the application, that is the sum of the slices he is operating, like keys and other data might contain facts on the customer allowing e.g. to analyze his business model or revenue system. Also end user specific data might be stored requiring the protection of the privacy of him. Isolation of the slices on the level of data access and information flows is due to these reasons required. This second dimension of isolation guarantees that the customer receives the agreed level of service he has paid for. It is important to understand that the isolation properties are only effective between the customers resp. their slices. The underlying system of the node is always able to manipulate the processes and data in the slices running on top. Due to this each customer has to trust the owner of the nodes (e.g. the ISP). Nevertheless the customer requires a mechanism to get proof that the node system is working as agreed in the contracts. By this he is able to (i) testify the integrity of his application and (ii) that this reported state is not influenced by the node.

To prevent malicious nodes to attack the P2P network each node needs to check the authenticity of the incoming requests. Authenticity of a particular request is satisfied for one node N if N can deduce that this request has occurred from its knowledge about the global behavior of the distribution network and the view of the current behavior. The identity of a node issuing requests to other nodes that lead to actions is the anchor for the authenticity of this action. Therefore, this identity needs to be protected against forgery or manipulation even from an attacker who has physical access. Also the authenticity of the state the corresponding node is in needs to be reviewed in the decision of on the authenticity of the corresponding action.

All nodes are autonomous operating devices located in the households of the end users. They are not under the direct physical control of the owner but are considered as fully controlled by him. As the end user has a big interest to access data on the node nodes are facing different kinds of attacks that can be categorized in three attack vectors. First, undetected modification by direct physical attacks should not be possible. Second, attacks by external attackers to modify the content. Third, the introduction of malicious or manipulated content by e.g. the legitimate end user with the aim to distribute illegal content should be prevented. A measurement system for the analysis of the actual state of the running software is required first to allow the individual node to detect tampering and second to report this state securely to the connected nodes. If the node is tampered it has to be assured that relevant information stored at the device is not accessible. Communication has to assure that based on this reported state only trustworthy nodes are part of the network.

The protection of the integrity and confidentiality is part of the customer applications as the requirements depend much on the content actually distributed. Content is distributed in data chunks between the involved nodes arriving from different nodes at the target node and is then assembled to the data stream presented to the end user or redistributed to other nodes. Thus, integrity protection offered by the architecture can only look at the individual data packages exchanged between the nodes but not on the whole content. Confidentiality as seen by the architecture is concerned with the isolation of the slices and with this preventing unwanted interaction between customers. Confidentiality on the data level has to be defined and implemented by the customer in his application.

3. Architecture and Functionalities

In this chapter first the high level architecture for the NaDa infrastructure is presented and then certain security functions presented required to establish the required level of security. In this chapter we assume a well developed node system and do not specify details of the internal design.

The abstract idea of the NaDa infrastructure is that a cloud of nodes operated and owned by an ISP delivers the possibility to host different slices provided by the customers. The customer on the other hand has the aim to establish an application that can be provided to his end users. Supporting these requirements (i) a node management is required to deploy and maintain slices and (ii) to protect and configure these slices. The customer has to control the functionality of his application and therefore needs certain management structures in his application. These two different management domains are considered as independent.

Aside of the management certain functionalities to identify the nodes and the slices need to be provided to (i) securely distinguish individual nodes and (ii) make sure that only nodes belonging to the P2P network are allowed to access data. This abstract definition of the NaDa model omits details on how and when to authenticate and other functional specifics.

3.1 High level architecture

The architecture of the NaDa infrastructure is based on the P2P paradigm allowing for a highly available infrastructure by distributing some of the core functionalities into the nodes installed. The resulting high level architecture is depicted in Figure 1 and consists of the node and its supporting infrastructure. Mainly two players' components are shown there. The customer is represented by its slice and the corresponding centralized application tracker. as well as its support by the customers Identity Provider and the store of content offered by the customer. In Figure 1 only one customer is shown for the sake of simplicity. Additional customers are duplicating the customer related components.

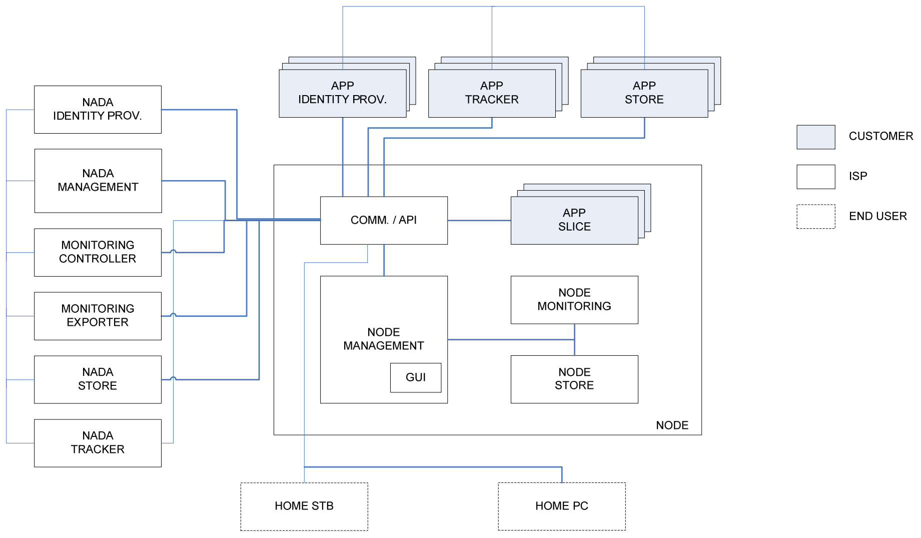

Figure 1 High level architecture for the distribution of content

The customers slice represents its main functionality that is offered to the end user through a graphical user interface (GUI) that provides for a common interface for all customers. All content of the particular customer is managed and maybe also charged by the slice. Its interaction with other slices of the same customer or its supporting infrastructure is controlled according to policies by

P2P and communication service (COMM/API). The corresponding application tracker that is specific for each customer defines the customers overlay network. In a simple approach this service is centralized. More complex applications may implement a combination of distributed and centralized tracking. Such a module is then located within the central P2P service.

The slice is supported by the application store that provides the content offered by the customer to the end users. The implementation of such a store could be done e.g. by specialized nodes injecting the content according to certain policies. Second, the customer operates an Application Identity Provider (AIDP) responsible for registering and managing the slices belonging to the application.

The ISP as the owner of the service offered to its customers operates the COMM/API on each node that implements the head unit in the communication with the node. The COMM/API intercepts all communications from and to a slice. This is used both for monitoring and filtering (e.g., firewalling) and may also provide communication primitives (e.g., P2P transfers, multicast, etc.) to slices and the node management module. Different policies can be applied within this service to prevent unwanted communication between slices or nodes. In this service also security mechanisms are in place ensuring that only trustworthy devices are accepted as communication partners. Through this service the application tracker of the individual customers are gathering their information on other slices of the same customer.

Monitoring is an essential part of the NaDa architecture as this component provides the data that allows for the resource efficient usage of the resources provided by the ISP. Also it is required to check for the conformance to the SLA with the respective customers. Due to these tasks the application tracker can be considered as part of the monitoring capabilities shown in section 3.3. Monitoring is implemented by two modules. In each node the node monitoring (NM) provides the necessary measurement values later stored in the nada store. On the infrastructure side the monitoring controller handles the NM module of all nodes and defines policies accordingly. These policies are used by the node management to configure the local node monitoring. Monitoring data is stored in the NaDa store (and possibly node stores). The monitoring exporter (ME) receives queries from the app tracker (AT) for monitoring information, and replies with properly formatted and filtered data. This functionality may be implemented also in a distributed approach to reduce the communication overhead. In the distributed implementation a local exporter situated in each node receives the queries of the AT and responses accordingly w.r.t. certain policies managing the interaction of the node and protecting the privacy of the different stakeholders. In this case AT needs also to be implemented distributed.

The application store provides for slices of customers and relevant system updates and is typically implemented as an external CDN. This functionality may also be provided as a distributed service to gain benefits in terms of resource efficiency and availability from the P2P approach. Its content is managed by the nodes management to ensure that the store consumes only the necessary amount of space. The software stored is defined and provided by the software management as an external service operated by the ISP. The propagation of slices to the individual node is based on a signaling mechanism and the NaDa tracker service.

The NaDa store is a service (typically but not necessarily centralized) that stores NaDa-related information (e.g., slice images, monitoring data). The node store consists of local storage space used by local slices and NaDa management services. It is to be noted that this service handles privacy relevant data and also business relevant information. Access to the data stored in this system is protected against access by appropriate means.

The node management module performs based on various policies the software update, slice instantiation, incentive measurement, and measurement of different values of the operation. These measurements are provided to the slices using the communication service. From the security perspective updates of the existing system and the rollout of new slices is of a special interest as by these operations the system configuration is changed. To comply with the identified security requirements all the packages have to be authenticated by the ISP to prove the origin of the software. It is also advisable to only accept software only from hosts that are recognized as trustworthy to provide a high protection against denial of service attacks. The responsibility for this is also in the domain of the NaDa store. All consequences due to changes to the state of the node are also to be handled by the node management in cooperation with the NaDa IDP.

Similar to the AIDP of the customer the NaDa IDP is responsible to identify a certain device and to issue credentials for all nodes involved in the P2P network stating that they belong to it. Part of this process is the verification of the authenticity of the state meaning that the software on the node is not tampered or otherwise harmed. This functionality is provided by a central entity from the ISP. Nevertheless, it is possible to allow also nodes to function as representatives for the ISP and by this distributing the functionality partly in the network. It is required that the AIDP is aware of the NaDa IDP to check certificates issued by the IDP.

Additionally a central management is established using existing protocols like the TR 069 allowing for a direct interaction with each individual device. It needs to cooperate with the NaDa IDP to securely identify the nodes he is communication with. Using this system the ISP is able to directly interact and change nodes. Again security means are required to cope with changes to the state of the system.

3.2 Use Case: Slice Installation and Content Distribution

Based on the presented architecture this chapter shows is structured use case how the different components are interacting. The selected business case is Video on Demand implemented based on the NaDa architecture. We assume the preconditions that communication between nodes in the overlay network is secured, by usage of trusted tickets [5]. Additionally, only nodes with an attestation inside the overlay network can participate in the communication. Therefore, nodes powered on are "logged on" to the ISP. The download definition file (DD file) includes the used APP tracker to get certain content and the partitioning of the content.

We differentiate between the Installation of the customer resources, the actual content distribution, accounting, and the end user requirements.

Installation of Customer Resources

The customer has to implement his APP Tracker. Optionally APP identity provider and APP store using the COMM/API defined by NADA can be implemented by the customer. These services must be available, when the ISP initiates the installation of the APP slices. Since operating system virtualization is used to install APP slices, the customer has to create an image (e.g. XEN) with his application using the NADA COM/API to implement the end user interface an the P2P protocol. To allow for operation the customer must provide resources for initial access to content by his APP slices via his APP tracker.

The image created by the customer must be certified by the ISP and the customer NADA management provides these images for slices on the NADA store. The ISP also defines appropriate resources and bandwidth for those slices as policies that are later on enforced by the respective NaDa components. As Slices have to be distributed the resource location for the DD file of slice images must be forwarded to all nodes logged on to the ISP. Nodes which receive this notification initiate the P2P protocol by sending a request with the downloaded DD file to NADA tracker. The P2P protocol to download the slice and to save it in the node store will be processed with NADA tracker and nodes in the overlay net announced by the NADA tracker.

The fingerprint of the complete downloaded slice image will be compared with the value from NADA store, to assure that a valid current version of the image is downloaded. The virtual hardware trust anchor of the slice image is initialized. Current time is stored in the trust anchor to enable the slice to generate timestamps certified by a trusted third party (e.g. the ISP). Finally, the APP slice registers itself to the APP tracker and is ready to receive DD files from the APP tracker and for end user access.

Content Distribution

After receiving a DD file the APP slice starts the P2P protocol for downloading the content defined in the respective DD file from the APP slices announced by the APP tracker. Data is stored encrypted in the NODE store and the decryption is possible for the APP slice only. Separation of the communication is assured by the policy provided by the COMM/API: “Only connections between slices with identical customer are permitted”

The APP tracker (i) distributes DD files according to expected demand and results of queries to the monitoring controller of the ISP, (ii) accepts queries for content and responds with the appropriate DD files and (iii) transmits peer lists to clients and collects state information of the APP slices.

Accounting

Each node keeps a log file with timestamps to prove the uptime of the individual node to the ISP. This file should be delivered to the ISP in defined time intervals and provides the base for incentives schemes. This is required to enforce the availability of the individual nodes.

End User Requests

The COMM/API signs and stores user requests in the node store, to assure non repudiation by a trusted third party. The corresponding DD file is received from the APP tracker and the P2P protocol is initiated. After the complete content is available or it is possible to start a stream data will be decrypted and prepared for delivery to the end user (e.g. protected by using watermarking techniques). Delivery takes place by using the COMM/API. COMM/API provides a log file stored in NODE store where parts of the content with a certain size delivered to the user are logged, also to assure non repudiation and to allow for accounting.

3.3 Monitoring Service Architecture

Applications need information about the status of nodes and paths between nodes. For instance, VoD needs to determine the best nodes to serve content to a given client; and gaming needs to track the quality of the connectivity among players. The notion of “best” and “quality” depends on measurements of the current status of nodes (e.g., CPU load) and of paths (e.g., available bandwidth and delay). Instead of having applications perform their own measurements; the NaDa monitoring services will keep up-to-date status of nodes and paths in the NaDa platform. Applications can then query these properties. The approach of combining all measurement efforts in the platform has the advantage of avoiding redundant measurements that could overload the platform.

Fig. 2 presents the functional architecture of the NaDa monitoring services. There are two main functions performed by the monitoring service (presented inside the dashed box). *Data collection* performs measurements and stores them at a management information base (MIB), i.e., a database of measurements. *Data export* retrieves measurements from the MIB to answers queries from authorized application controllers and from the NaDa management module. Next, we detail each of these functions.

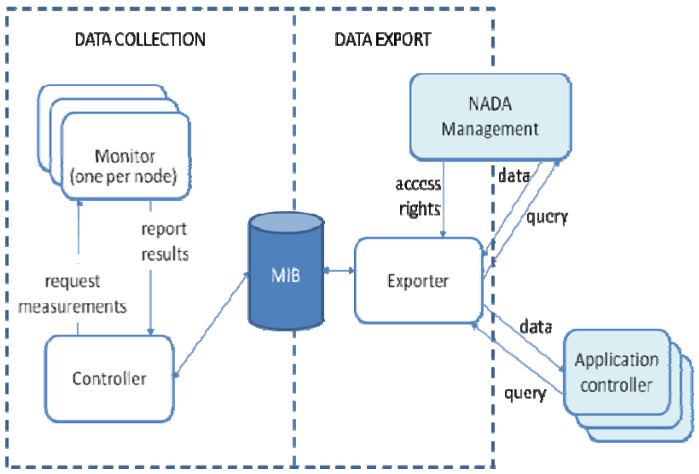

Figure 2: NaDa monitoring service architecture.

Each node has a *monitor*, which is responsible to trigger active measurements from this node and to passively monitor slices running in the node. The *controller* orchestrates all data collection. It requests measurements from monitors, so that the MIB is up to date. Fig. 2 depicts one controller and one MIB. However, these functions can be centralized at one monitoring server or distributed in multiple servers or platform nodes. The details of the implementation of these functions are part of the research of this project.

NaDa will monitor (i) per slice the uptime and CPU, memory, and storage (RAM and hard disk) utilization, (ii) per node the Link utilization, power consumption, and disk (I/O) bandwidth; the set of active applications, and (iii) per network path between pairs of nodes the reachability, path topology, capacity, delay, jitter, loss, and available bandwidth. These properties are essential for the management of the NaDa platform and for the target applications of VoD and gaming.

The creation of a MIB to represent these properties poses a number of challenges. First, some of these properties (like link utilization, delay, jitter, loss, and available bandwidth) are highly variable over time. We need to specify metrics to summarize measurements of such variable properties that are meaningful to all applications. Second, applications need up to date information. However, updating the MIB too frequently could represent a significant overhead. We need to find the good comprise between freshness of information and measurement overhead. Finally, the large scale of the NaDa platform is problematic. The NaDa platform should have thousands of nodes and the monitoring module has to report properties about paths between every pair of nodes. We need to decide whether to represent every path in the MIB or to summarize this information (by using a virtual coordinates, for instance). WP2 will address all these challenges.

The *ME* receives queries from application controllers and from the NaDa management module and responds with the requested properties. Applications can only access measured properties for which they have the right credentials. The exporter is responsible to authenticate every query and filter the results. Applications only have rights to query (i) properties of their own slices and (ii) neutral properties of nodes where they have a slice and paths between these nodes. A *neutral property* is one that does not reveal information about other applications running at the NaDa platform. For instance, the reachability between two nodes is neutral, because it does not depend on the applications running on these nodes; whereas the link utilization of a node depends on the activity of the applications in the node.

Application controllers communicate with the *ME* using a query interface. The query interface allows to (i) query properties of a single element and sets of elements: If application controllers query the property of a single element (for instance, the link utilization of a node, the available bandwidth of a path, or the power consumption of a node), then the exporter just answers with the appropriate value. However, if the application controller needs properties of all its slices or of paths among all its nodes, then querying each property separately is cumbersome. Instead, the exporter allows select and project operations that will return a database with only the requested properties. (ii) The query interface allows users to define the resolution of the results. Possible choices are to report the instantaneous value of a property, statistics, or a qualitative metric (for instance, the application controller can define a range of values that are good and bad). (iii) Application controllers can request that the monitoring service track the status of an element and raise an alarm under a given condition.

3.4 Security primitives

As required in the security analysis functionalities concerning identity and authenticity of the actual state have to be implemented in this architecture providing an appropriate level of assurance. As software is also prone to certain attacks both methods have to build up on hardware protected capabilities as provided by Trusted Computing [9]. The Trusted Platform Module (TPM) offers by its hardware design the required level of assurance regarding the core functionalities to allow for a tamper proof identity. In various systems identities are represented using tokens containing information on the person or device and stating that a certain entity belongs to the group accepted. A well known example in the network domain is the Kerberos protocol [4]. Preventing duplication and extraction these tokens need to be tightly coupled with the node. Trusted Tickets, as presented e.g. in [6] can be used to grant an individual node access to the overlay network and the services offered within. These tickets provide the binding between the identity of a TPM that is protected by the hardware means of it and the assigned identity in the network.

The security of the identity is only depending on the security assertions of the TPM. Providing proof on the status of the device needs a mechanism to securely communicate status information to the other nodes. Remote Attestation as presented above enables each pair of nodes to communicate its state to the other. Each on turn than has to decide if he is accepting the provided proof and allows for communication. This mechanism is part of the P2P and communication service as it is part of the node. The state of the individual node corresponds with the software running on it. This includes the BIOS, boot loader, and the actual OS. Applying updates as well as malicious changes to the node lead to a change of the status. It is to be noted that the attestation process only covers static behavior as it is defined by the executables. Dynamic changes e.g. due to a buffer overflow can not be detected by this approach. Therefore, efficient protocols are required for securing the updates in terms of authenticity and integrity and after the updates are applied to check for the state.

Existing tickets covering the state of the node are rendered useless after an update due to the change of the state. Therefore the node requires new credentials issued by the NaDa IDP component. To shape this process according the availability requirement on the one hand and the performance requirement on the other it is reasonable to integrate the reissuing of credentials as part of the software roll out in a distributed approach. The concept of ISP representatives as part of each node can be applied at this point.

Another functionality required is the binding of secrets to a certain state of the system so that if the system is altered these secrets are no longer accessible and therefore rendered useless for an attacker. This is also supported by trusted computing as shown in chapter 3. Based on this functional building block a customer or ISP is able to send data bound to a certain desired state of the node. This can be used in different protocols for e.g. update or software roll-out schemes. Within the slices of the customers data can be bound to the state of the underlying node system by using e.g. Trusted Computing [7,8,9]. This allows for the protection of credentials used by the customers and also from the ISP.

Beside the protection of keys it is required to protect also resources used by the customers. This is done in general by isolation resp. virtualization technologies as for example XEN, Linux VServer, or KVM. An isolated or virtualized system as it is used in a slice only sees what is offered to him by the host system. The host is in the position to interfere with all data and processes owned by the slice. Remote Attestation requires therefore (i) the proof of authenticity of the state of the slice and (ii) the same proof also from the host system. At some point in the business model content is delivered to the end user. At this point the content is transferred to systems not within the system operated by the ISP and therefore not part of the security perimeter established by the nodes and the used protocols. To prevent misuse of delivered content different schemes can be applied. Watermarking or DRM (e.g. OMA DRM) methods are available for this purpose. Applying these schemes has to be done by the node as the content is delivered to the node by the P2P protocols preventing the application of protecting schemes by central servers as there is no direct delivery of the content.

4. Conclusion

This paper presents the architecture and the security functionalities required in a novel distribution scheme for virtual goods as they are identified by the Nanodatacenters project. The special challenge herby is that well known mechanisms like per node encryption are no longer applicable. Therefore several aspects of distribution protocols and architectures need to be discussed. Present work covers the development of appropriate P2P extensions for adding strong identities and remote attestation. Especially the efficiency of these protocols in terms of latencies is crucial. Also the combination of Trusted Computing and virtualization technologies is part of the research activities. Concepts to propagate trust between the devices are the central aim as the customers require good protection for their goods.

All the discussed security techniques are supporting the business case of a distributed platform for virtual goods. The privacy of both customers as well as end users is of interest as it is crucial for the acceptance by them. Due to this also the measurement architecture has to be covered by the security architecture.

References

[1] B. KRISHNAMURTHY, C. WILLS, AND Y. ZHANG. On the use and performance of content distribution networks. In Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement, pp 169-182. ACM New York, NY, USA, 2001.

[2] I. STOICA, R. MORRIS, D. KARGER, M. KAASHOEK, AND H. BALAKRISHNAN. CHORD: A scalable peer-to-peer lookup service for internet applications. In Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, pp 149-160. ACM New York, NY, USA, 2001.

[3] N. KUNTZE AND A. U. SCHMIDT. HANDBOOK OF RESEARCH ON SECURE MULTIMEDIA DISTRIBUTION, Trust in the Calue-Creation Chain of Multimedia Goods. IGI Global Publishing, 2009.

[4] STEINER, J., NEUMAN, C., AND SCHILLER, J. 1988. Kerberos: An Authentication Service for Open Network Systems. In Proc. Winter USENIX Conference. Dallas).

[5] KUNTZE, N., LEICHER, A., AND SCHMIDT, A. 2009. Implementation of a trusted ticket system. In Proceedings of the IFIP Security 2009.

[6] BRETT, A. AND LEICHER, A. 2009. Ethemba trusted host environment mainly based on attestation.

[7] TRUSTED COMPUTING GROUP, INFRASTRUCTURE WORKING GROUP: Architecture Part II – Integrity Management, Version 1.0 (2006).

[8] TRUSTED COMPUTING GROUP: TCG Specification - Architecture Overview, Version 1.2 (2006).

[9] MITCHELL, C.: Trusted Computing. Institution of Engineering and Technology (2005)