Trusted Ticket Systems and Applications

Nicolai Kuntze[1] and Andreas U. Schmidt[1]

Fraunhofer–Institute for Secure Information Technology SIT
Rheinstraße 75, 64295 Darmstadt, Germany
`{andreas.u.schmidt,nicolai.kuntze}@sit.fraunhofer.de`

Summary. Trusted Computing is a security base technology that will perhaps be ubiquitous in a few years in personal computers and mobile devices alike. Despite its neutrality with respect to applications, it has raised some privacy concerns. We show that trusted computing can be applied for service access control in a manner protecting users' privacy. We construct a ticket system — a concept which is at the heart of Identity Management — relying solely on the capabilities of the trusted platform module and the standards specified by the Trusted Computing Group. Two examples show how it can be used for pseudonymous and protected service access.

1 Introduction

In a ticket based authentication and authorisation protocols like Kerberos [1] software tokens are used to prove the identity of a single entity. Based on this tokens access to certain systems is restricted to entities producing appropriate tokens. Additionally data embodied in the token can be used to implement also an authorisation control enabling a token based access control scheme beside the mere authentication. These tokens are an electronic analog to physical tickets. They can have a limited validity period ore be used a specified number of times.

For the adept, some base concepts of Trusted Computing (TC) look very similar to Identity Management (IDM). We exploit the analogy between TC and IDM and construct a ticket system using TC functionality, thus obtaining a cornerstone of IDM systems. The applicability of such a *trusted ticket system* is shown in the context of a reputation system and a push service.

Section 2 provides necessary background on TC. Section 3, developing the trusted ticket system proper is subdivided in 3.1, explaining how to use Attestation Identity Keys (AIKs) for the realisation of tickets, and 3.2 detailing the proceedings for the acquisition and redemption of the latter. Section 3.3 outlines a general service access architecture utilising trusted tickets, with a high degree of separation of duties, providing for pseudonymity, accountability, and charging functionality. Section 4 embeds trusted ticket systems in the two mentioned application contexts and discusses resulting benefits. Conclusions are drawn in Section 5.

2 Trusted Computing Essentials

Trusted computing uses a hardware anchor as a root of trust and is now entering the mobile domain with the aim to provide a standardised security infrastructure. Trust in the context of TC means (as defined by the Trusted Computing Group, TCG) that an entity always behaves in the expected manner for the intended purpose. The trust anchor, called Trusted Platform Module (TPM), offers various functions related to security.

Each TPM is bound to a certain environment and together they form a trusted platform (TP) from which the TPM cannot be removed. Through the TPM the TP gains a cryptographic engine and a protected storage. Each physical instantiation of a TPM has a unique identity by an Endorsement Key (EK) which is created at manufacture time. This key is used as a base for secure transactions as the Endorsement Key Credential (EKC) asserts that the holder of the private portion of the EK is a TPM conforming to the TCG specification. The EKC is issued as well at production time and the private part of the key pair does not leave the TPM. There are other credentials specified by the TCG which are stating the conformance of the TPM and the platform for instance the so called platform credential. Before a TPM can be used a take ownership procedure must be performed in which the usage of the TPM is bound to a certain user. The following technical details are taken from [2].

The TPM is equipped with a physical random number generator, and a key generation component which creates RSA key pairs. The key generator is designed as a protected capability and the created private keys are kept in a shielded capability (a protected storage space *inside* the TPM).

The TPM possesses so-called shielded capabilities protecting internal data structures by controlling their use. Three of them are essential for applications. First, *key creation and management*, second the ability to create a *trust measurement* which can be used to assert a certain state toward a, remote party, and finally methods, which we call *sealing*, to protect arbitrary data by binding it (in TCG nomenclature) to TP states and TPM keys.

For the TPM to issue an assertion about the system state, a process called *attestation*, two protocols are available. As the uniqueness of every TPM leads to privacy concerns, they provide pseudonymity, respectively, anonymity. Both existing attestation protocols rest on Attestation Identity Keys (AIKs) which are placeholders for the EK. An AIK is a 1024 bit RSA key the private portion of which is also sealed inside the TPM. The simpler protocol of Remote Attestation (RA) offers pseudonymity by introducing a trusted third party, the privacy CA (PCA, see [3]), which issues a credential stating that the respective AIK is generated by a sound TPM within a valid platform. The system state is measured by a reporting process with the TPM as its central reporting authority receiving the measurement values and calculating a unique representation of the state using hash values. For this the TPM has several Platform Configuration registers (PCR). Beginning with the system boot each component reports a measurement value, e.g., a hash value over the BIOS, to the TPM and stores it in
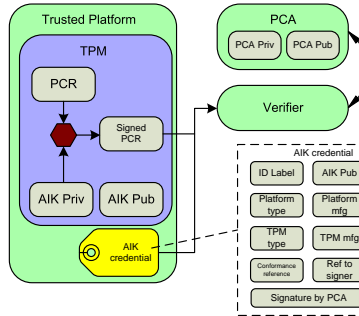
Fig. 1. Remote Attestation process.

a log file. During RA the communication partner which acts as verifier receives this log file and the corresponding PCR value. The verifier can then decide if the device is in a configuration which is trustworthy from his perspective. Apart from RA, the TCG has defined Direct Anonymous Attestation. This involved protocol is based on a zero knowledge proof but due to certain constraints of the hardware it is not implemented in current TPMs.

AIKs are crucial for applications since they can not only be used, according to TCG standards, to attest the origin and authenticity of a trust measurement, but also to authenticate other keys generated by the TPM. Before an AIK can testify the authenticity of any data, a PCA has to issue a credential for it. This credential together with the AIK can therefore be used as an identity for this platform. The protocol for issuing this credential consists in three basic steps. First, the TPM generates an RSA key pair by performing the `TPM_MakeIdentity` command. The resulting public key together with certain credentials identifying the platform is then transferred to the PCA. Second, the PCA verifies the correctness of the produced credentials and the AIK signature. If they are valid the PCA creates the AIK credential which contains an identity label, the AIK public key, and information about the TPM and the platform. A special structure containing the AIK credential is created which is used in step three to activate the AIK by executing the `TPM_ActivateIdentity` command. So far, the TCG-specified protocol is not completely secure, since between steps two and three, some kind of handshake between PCA and platform is missing. The existing protocol could sensibly be enhanced by a challenge/response part to verify the link between the credentials offered in step one and used in step two, and the issuing TPM. The remote attestation process is shown in figure 1.

Beside the attestation methods TC offers a concept to bind blobs of data to a single instantiation and state of a TPM. The `TPM_unbind` operation takes the data blob that is the result of a `Tspi_Data_Bind` command and decrypts it for export to the user. The caller must authorise the use of the key that will decrypt the incoming blob. In consequence this data blob is only accessible if the platform is in the namely state which is associated with the respective PCR value.

A mobile version of the TPM is currently being defined by the TCG's Mobile Phone Working Group [4]. This so-called Mobile Trusted Module (MTM) differs significantly from the TPM of the PC world and is in fact more powerful in some respects. In particular, it contains a built-in verifier for attestation requests, substituting partly for an external PCA. Both TPM and MTM are a solid basis for application architectures and scenarios. Trusted Computing not only affects the world of networked PCs but will also heavily impact the mobile industry. Accordingly, one of our application scenarios see section 4.2 is a mobile one. The concept for ticket system we present in section 3 is in fact network agnostic and can be applied to the Internet as well the mobile world. Further mobile scenarios can be found in [6, 7].

3 A TC-based Ticket System

The basic idea is to establish a (pseudonymous) ticket system using the identities embodied in the PCA-certified AIKs. Specific about our design is the tickets are generated locally on the (mobile) device of the user. Ticket acquisition an redemption rests solely on trusted computing methods implemented in the TPM chip embedded in the users platform. We first describe how AIKs can be turned into tickets that can be used in a ticket-based service access or identity management architecture, and then develop the processes for their acquisition and redemption.

3.1 AIKs as Tickets

For security considerations the TPM restricts the usage of AIKs. It is not possible to use AIKs as signing keys for arbitrary data and in particular to establish tickets in that way. It is therefore necessary to employ an indirection using a TPM generated signing key and certify this key by signing it with an AIK — *viz certify* it in the parlance of the TCG. Creation of a key is done by executing the `TPM_CMK_CreateKey` command, which returns an asymmetric key pair where the private portion is encrypted by the TPM for use within the TPM only. The resulting key pair is loaded into the TPM by `TPM_LoadKey` and thereafter certified by `TPM_CertifyKey`. By certifying a specific key the TPM makes the statement that "this key is held in a TPM-shielded location, and it will never be revealed". For this statement to have veracity, a challenger or verifier must trust the policies used by the entity that issued the identity and the maintenance policy of the TPM manufacturer.

This indirection creates to each AIK a certified key (by the namely AIK) that can be used for signing data, in particular the payload data of a ticket to be submitted to, and accepted by, a service. We call this key pair the *certified signing key* (CSK). CSK, AIK, together with a certificate by the PCA (see below) attesting the validity of that AIK, are the ingredients that realise a ticket for a single operation, e.g., a service access.

3.2 Ticket Acquisition and Redemption

Tickets are acquired by a *trusted agent* (TA), i.e., the user of a ticket system and associated services operating with his trusted platform, from the PCA. They are then redeemed at the *(ticket) receiving system* (RS). In both processes, a *charging provider* (CP) may occur as a third party, depending on application architectures. We now describe how these operations proceed.

Note that we do not distinguish between public and private key portions of a certificate establishing a credential. As a notation, the credential of some certified entity Cert(*entity, certificate*) means the union of the public key Pub(*certificate*) and the entity signed with the certificate's private key, $entity_{\mathrm{Priv}(certificate)}$. Verifying a credential means to check this digital signature.

An interesting option is that the credentials issued by the PCA for a AIKs can be designed as *group credentials*, i.e., they do not identify a single AIK *viz* ticket but rather its price or value group g chosen from a predetermined set indexed by the natural numbers $g \in \{1, \ldots, G\}$. The group replaces an individual identity of a platform and many TAs will get the same group certificate. Only the PCA can potentially resolve the individual identity of a platform. This allows for the combination of a value proposition with privacy protection, as these groups are used to implement price and value discrimination of tickets. Note that the PCA is free in the choice of methods to implement group certificates. This could be done by simply using the same key pair for the group or by the existing, sophisticated group signature schemes [5].

If a TA wants to acquire a rating ticket from group g, he first generates an AIK using the `TPM_MakeIdentity` command. Next, TA requests from the PCA a credential for this AIK, belonging to group g, by sending AIK, group identifier and supplementary data as required by TCG protocols, to the PCA. The PCA now knows the identity of the TA. This can be used to perform a charging for the ticket, either by contacting CP or by the PCA itself (how charging actually works is not in the scope of this paper). It is important that, at this stage, an authorisation decision on the ticket generation can be made by the PCA, for instance to blacklist misbehaving participants. If the authorisation succeeds (and not earlier, to save bandwidth and resources), the PCA performs a handshake operation with the TA to ensure that the AIK has actually been generated by the particular TPM in question. Upon success, the PCA generates the credential Cert(AIK, g) certifying that the AIK belongs to group g. The credential is transferred back to TA, where finally the `TPM_ActivateIdentity` command is executed to enable subsequent usage of this AIK. The process is shown on the left hand side of Figure 2.

Redeeming a ticket is now very simple, as shown on the right hand side of Figure 2. TA has first to generate a CSK, i.e., a public/private key pair and the credential Cert(CSK, AIK) for it according to the process described in Section 3.1. He then signs a certain payload, P, e.g., describing a service request, with CSK to obtain Cert(P, CSK). The payload and the credential chain Cert(P, CSK), Cert(CSK, AIK), Cert(AIK, g) is then transferred from
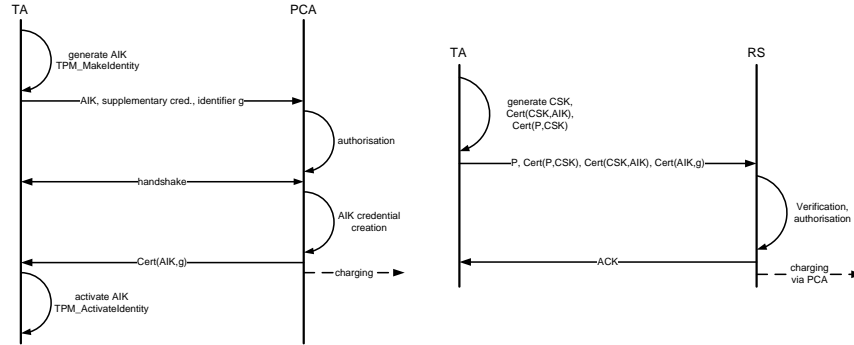
Fig. 2. Ticket acquisition (left) and redemption (right) processes.

TA to RS and this set of data embodies the ticket proper (we do not discuss a particular data format for the ticket). RS verifies this chain and makes an authorisation decision, for instance to implement a protection against multiple spending. Finally, RS acknowledges receipt of P and optionally initiates another charging operation (ex post charging) via PCA.

3.3 A Generic Architecture for Service Access

The embedding of the described ticket acquisition and redemption into an application system and business context offers many variants. A very basic scenario is shown in Figure 3. Here, a trusted agent (user) would like to access some service, and buys a ticket of a certain value from the PCA. The ticket belongs to a certain group which can represent statements such as "for usage with Service n", and a certain value, monetary or intangible, e.g., in a rebate scheme. The user then issues a service request as payload in the ticket redemption toward RS. The TA pays for the ticket at the CP at the time of redemption of the ticket and the CP distributes revenue shares between himself, PCA, and RS, according to service level agreements. RS, in turn, remunerates the service (an acknowledgement of service processing is omitted for simplicity).

This realises an access control scheme to multiple services mediated by PCA and RS, yielding three essential benefits: 1. non-repudiation by the chain of credentials, 2. accountability by resolution of the TA's identity through PCA, and 3. pseudonymity by separation of duties. The PCA/RS combination plays a very central role for the control of identities embodied in the pseudonymous tickets that PCA issues. It is in fact an embodiment of the role of an identity provider in a ticket-based identity management (IDM) system. That TC can be used to model IDM was outlined in [6, 7], and is presented here for the first time in detail.

Though the separation of duties between PCA and CP allows in principle even for anonymity of the person using a TA, since only upon charging this person must be identified by credit card account or other means, this may
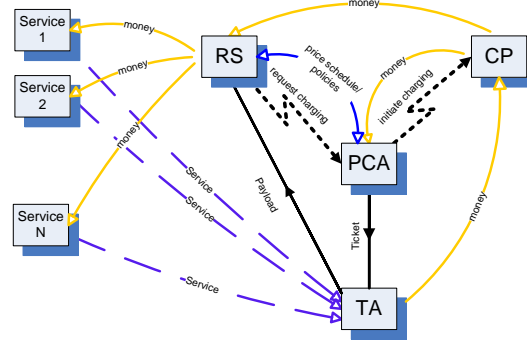
Fig. 3. A generic architecture using TC-based tickets.

not be the best option. In fact this would loose some accountability of the TA users. While RS may be able to obtain personal identities from PCA if pertinent contractual relationships are in place, e.g., if fraud by a TA user is suspected, data protection regulations may prevent a CP from unveiling personal identities. A second role played by the PCA is the initiation of charging. With respect to the revenues from ticket sales, a natural approach is a sharing between RS and PCA (and CP for its service). RS and PCA negotiate and implement policies for authorisation within the ticket acquisition and redemption processes, e.g., to prevent double spending or to blacklist misbehaving users. In collaboration between PCA and RS, practically any price schedule can be realised.

This architecture naturally extends to an arbitrary number of receiving systems to which PCA offers ticket management, and in particular pricing, as services. A further extension would be to let TA express values of tickets by using different (groups of) CSKs. In this way tickets can be associated with additional certified attributes, e.g., priorities.

3.4 Security and Privacy

The presented method for the management of tickets provides for perfect pseudonymity of the participants toward the system. In fact, only PCA is able to de-anonymise users.

Note that for the namely reason only PCA can initiate a charging, since only he knows (or is able to know) the identity of a TA and can link it to the identity of the corresponding participant. To keep this pseudonymity strength, it is essential that our concept relies only on genuine TPM functionality, and in particular avoids the usage of trusted software. If there was a trusted software managing tickets in some way at the side of TA, then this software, and the state of the platform would have to be attested both in ticket acquisition and redemption. To this end the TC protocols for remote attestation transfers trust measurements and measurement logs to the corresponding verifier (PCA or RS in our case). These data can however — and this is a principal problem

with remote attestation — be used to individualise the trusted platform, if, as in the PC domain, the number of system states and different measurement logs created at boot time, is very large in relation to the number of users of a TC-based service. Besides, avoiding remote attestation saves bandwidth and resources consumption. Since in this case no trust can be laid in the TA for ticket management, some kind of double, or multiple, spending protection or usage authorisation is needed at RS upon ticket redemption.

On the other hand security necessitates additional means of protection of content in transit. In many applications, for instance if confidentiality of transported payload is a protection target, trusted software usage cannot be avoided We show in Section 4.2 how TC can be used to establish end-to-end protection for P, but this definitely requires trusted software clients at both ends.

4 Two Applications

4.1 Price Scheduling in Pseudonymous Rating Systems

This application has been outlined in [8]. Electronic market places for physical and information goods are increasingly occupied by self-organising communities. These market places exhibit the characteristics of the so-called long tail economy [9]. That is, the classical asymmetry between suppliers and consumers is lifted. Buyers and sellers are often even in numbers and may change their roles dynamically. Virtual, or physical, goods are offered in large numbers and diversity and with potentially small demand for each single one. Matchmaking and orientation of buyers is difficult in a long tail economy, long term relationships are hard to build, and trust between trade partners must be established somehow [10].

A common approach is to let market players themselves provide the necessary guidance. This is mostly embodied in reputation systems by which buyers and sellers rate each other and the goods sold, or recommendation systems, i.e., programs which attempt to predict items that a user may be interested in, given some information about the user's profile. Reputation systems, according to Paul Resnick *et al.* [11] "seek to establish the shadow of the future [the expectation of reciprocity or retaliation in future interactions, cf. [12]] to each transaction by creating an expectation that other people will look back on it". The goal is to establish a homogeneous market for honest participants. That community ratings (of goods) do in fact strongly influence buyer behaviour is shown empirically in [13].

Existing reputation systems are fragile, in that they can easily be distorted or abused even within the frame of laws governing them. 'Attacks' of this kind, though not proper attacks in the sense of information security, threaten the integrity — with respect to its purpose — of the informational content stored in the system. Dellarocas [14] classifies unfair behaviour into the categories 1. *Ballot stuffing*: A seller colludes with a group of buyers in order to be given unfairly

high ratings. 2. *Bad-mouthing*: Sellers and buyers collude to rate other sellers unfairly low to drive them out of the market. 3. *Negative discrimination*: Sellers provide good services only to a small, restricted group of buyers. 4. *Positive discrimination*: Sellers provide exceptionally good service to some buyers to improve their ratings. A situation of controlled anonymity in which the market place knows the identity of participants and keeps track of all transactions and ratings, but conceals the identity of buyers and sellers, is identified as essential to avoid unfair behaviour. For instance, anonymity is an effective protection against bad-mouthing, but cannot work for ballot stuffing as sellers can give hidden indications of their identities to colluders.

On the other hand, the best known individual attack on reputation systems uses Sybils to obtain a disproportionately large influence [15]. Friedman and Resnick [16] point to the general problem of 'cheapness' of pseudonyms in marketplaces and reputation systems, since with name changes dishonest players easily shed negative reputation, as corroborated theoretically in [17]. The paper [18] gives an explicit threshold for the transaction costs for reputations needed to avoid ballot stuffing. However, an indiscriminate pricing of identities for the submission of ratings poses an undesired entry deterrent. It seems therefore plausible that reputation systems should be based on pseudonyms which allow for a flexible forward pricing.

While related work addresses particular vulnerabilities [19], or proposes general frameworks to ensure accountability in reputation systems while maintaining anonymity [20, 21], we here propose a simple mechanism to introduce arbitrary costs for pseudonyms. The separation of duties between PCA and RS in our ticket system implements here precisely the controlled anonymity desired for reputation systems through the properties 1.–3. mentioned in Section 3.3.

Again, the generic ticket system can be embedded in various ways into a (commercial) reputation system. If a TA user wants to express a rating about another user (for example, a buyer about a seller, a seller about a buyer), he buys a *rating ticket* from the PCA. The group attribute of the ticket in this special context expresses a value proposition for the rating, e.g., an impact factor used by the rating system to calculate weighted overall ratings, as well as an attribution to a particular rating system. The user then formulates the rating and sends it to RS as ticket payload, and charging is executed. The result would be a rating statement about another participant of the rating system which is trustworthy, accountable, but protected as a pseudonym. This enables the resolution of one important problem in reputation systems, namely accountability of users, i.e., the possibility to trace back malicious ones and threaten them with consequences. Based on the trusted ticket system, price schedules can be adapted to the requirements of rating systems as laid out above. On the extreme ends of the spectrum are cost-free registration of ratings by PCA, ensuring only accountability, and increasing charges with the number of ratings (or, e.g., their frequency). Even reverse charging, i.e., paying incentives for ratings, e.g., such of good quality, is possible.

4.2 Content Protection for Push Services

Workers occupied with 'nomadic' tasks depend on infrastructures for easy and swift access to required data. E-Mail push services like RIM's Blackberry conquer this market with huge success, and aim at high availability and ease of use. Push services are characterised by the ability to notify end users of new content. For an e-mail service the end user device is activated by the mail server, gets new mail, and notifies the user. The basic method for this has been formulated for instance in the standards of the Open Mobile Alliance (OMA, see [23]). Some providers have extended their range to enable access to company databases and implement loosely coupled work-flows incorporating nomadic workers.

Due to the high value of the exchanged data these systems are threatened by, even professional, attackers, raising the requirement to protect the distribution of pushed data. Security concerns can be grouped in two main areas. First, data has to be protected in transit to the device, and confidentiality is to be maintained. Second, after the data is delivered it has to be protected against unauthorised access. The latter problem is of practical importance in use cases like the mentioned e-mail push, but also for SMS delivery, and other data synchronisation processes between a central data base and a mobile device. Current approaches to this challenge are predominantly using software tokens, e.g., PKCS#7, to secure message transport and storage. Such solutions however suffer from the drawback that an attacker can extract keys from memory during encryption or decryption of a data block. Smart cards are a more evolved approach, but in the mobile domain no standard has become prevalent.
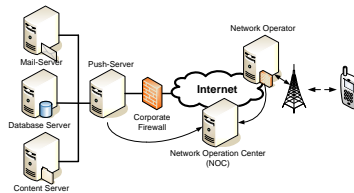

Fig. 4. A centralised architecture

Figure 4 illustrates the widely used, centralised push architecture. In its centre a Network Operation Center (NOC) performs all tasks regarding the communication to the mobile devices. The data destined for the mobile devices are stored in sources like mail servers. These sources excite the push server and deliver the data. Due to this activation the push server either requests a communication line to the mobile device managed by the NOC or delivers the data to the NOC which in turn stores the data until they are handed to the mobile device. From a company's view the management costs are low as there is no additional effort needed to maintain, e.g., a special firewall configuration. The decentralised counterpart includes direct communication between push server and mobile device, requiring access over, e.g., the Internet to servers behind a

company firewall, in turn necessitating special protection of the internal infrastructure. Taking a malicious service provider into consideration privacy concerns are added to the general ones regarding transport security. Potentially the NOC can access every message which is sent to a mobile device, allowing for information leakage. First and foremost the message content could be extracted and disclosed. Moreover, analysis of the collaboration between active users becomes possible. Both attacks have to be treated in the protocol design to enable end-to-end security and to conceal all sensitive information.
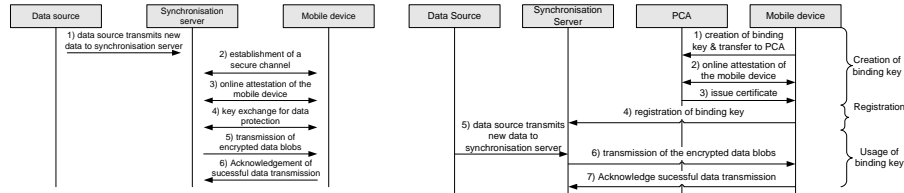

Fig. 5. Scenarios based on blob sealing (left) resp. key sealing (right)

Protection of content is a major use case for trusted computing [22]. We first present basic options to protect push content, and then describe the integration with th trusted ticket system of Section 3. Based on blob binding (see Section 2) the simplest scenario see the left diagram in Figure 5. First, the data source signals the synchronisation server (SyncS) which then locates the target device, establishes communication, and controls the synchronisation relying on, e.g., OMA data synchronisation. Channel security can be realised using Transport Layer Security. Remote attestation of the device and key exchange for payload encryption are performed. The data is transferred to the device and stored in sealed blobs, see [2, Chapter 12], only accessible in the determined state of this unique device. This approach suffers from latency produced by the Steps 2-4, in particular attestation creates computational load and produces some traffic.

An approach to meet this challenge is shown on the right hand side of Figure 5. SyncS encrypts the data with a public part of a key pair. The use of the corresponding private part is restricted by a PCR value. To grant trust in this public key the SyncS requires a certificate issued by a PCA, and a certified PCR value, signifying, e.g., the presence of an e-mail application in a certain configuration and a well defined environment. The PCA which certifies the platform keys can actually be used here to issue certificates augmented by the information of the PCR value. This variant of an AIK is further called a *binding key*. Step 1 transmits the public portion of a key pair to the designated PCA for content encryption keys. Remote attestation is performed and a certificate is created stating that the key originates from a TP and is usable if and only if the platform is in a certain state. This certificate is transmitted to the mobile device, and then, together with the binding key, to SyncS. Steps 1 to 4 only have to be performed once during roll out of the device or take ownership by its user.

In Step 5 the data is transmitted to SyncS, which can now encrypt the data with the binding key or with a hybrid scheme. Data transmission is executed in Steps 5–7. Note that the first presented method is much more flexible than the second one, since in the former the server can decide each time if the particular device can be considered as trustworthy.

The combination of the ticket system with the content protection system proceeds as follows. After it has been activated, the mobile device uses a ticket obtained from the PCA as a token to access SyncS which can either be the RS itself or an associated service. The ticket AIK cannot be used for data encryption, but the ticket payload can carry the mentioned public key for content encryption, *and* its certificate. The ticket PCA can act concurrently as the CA certifying the binding key or in separation from it. Ticket grouping or prioritising can be used advantageously in such a scheme, e.g., for attribution of bandwidth to a request and load balancing.

5 Conclusions

We have shown how to generate and use tickets based on TC, and have provided theoretical proof-of-concept in two independent application scenarios employing trusted tickets. What we have constructed is essentially a payment system with a trusted third party guaranteeing pseudonymity. It is therefore worthwhile to compare our method with the use case scenario "Mobile payment" of TCG's Mobile Phone Working Group Use Case Scenarios [22, Section 8]. There, the focus lies on device-side support of payment operations on a mobile phone which is turned into a trusted platform. This always involves a trusted software on the device which is not required in our approach. On the other hand this is only possible through the introduction of a trusted third party, the PCA with its extended duties. Thus we lack the universality of client-side solutions. Yet we have shown that a very simple ticket system with strong pseudonymity can be established resting solely on the most basic TPM functions.

It should be noted that our applications use TC in a way very different from Digital Rights Management (DRM), which is often considered as the sole use for TC. Both applications bind the economic value to a particular instantiation of the TPM. If this trust anchor breaks, only a limited damage can occur as the damage is restricted in space and time, e.g., to a single data synchronisation in a push service or submission of a single reputation. In contrast, if a single TPM in a DRM system breaks, the protected digital good can be converted into an unprotected version which can be freely distributed on a large scale, causing heavy monetary losses to its owner.

It is interesting to note that introducing TC yields a secondary user bound identification token. Due to the take ownership procedure of the TPM it is bound to a certain user. Therefore it is in its function very similar to a SIM as it is also possible to migrate the relevant parts from one TPM to the next.

One may ask whether two different identification tokens will survive in future TC-enabled mobile devices.

References

1. Massachusetts Institute of Technology: Kerberos: The Network Authentication Protocol. http://web.mit.edu/kerberos/ 1
2. Trusted Computing Group: TCG TPM specification version 1.2 revision 94. Technical report, TCG (2006) 2, 11
3. Trusted Computing Group: TCG Infrastructure Working Group Reference Architecture for Interoperability (Part I) Specification Version 1.0 Revision 1. Technical report, TCG (2005) 2
4. Trusted Computing Group: TCG Mobile Trusted Module Specification. Specification version 0.9 Revision 1. Technical report, TCG (2006) 4
5. Chaum, D., van Heyst, E.: Group signatures. In Davies, D., ed.: Advances in Cryptology - EUROCRYPT '91. Volume 547 of Lecture Notes in Computer Science, Berlin, Heidelberg, Springer-Verlag (1991) 257–265 5
6. Kuntze, N., Schmidt, A.U.: Transitive trust in mobile scenarios. In Müller, G., ed.: Proceedings of the International Conference on Emerging Trends in Information and Communication Security (ETRICS 2006). Volume 3995 of Lecture Notes in Computer Science (LNCS), Springer-Verlag (2006) 73–85 4, 6
7. Kuntze, N., Schmidt, A.U.: Trusted computing in mobile action. In Venter, H.S., Eloff, J.H.P., Labuschagne, L., Eloff, M.M., eds.: Proceedings of the Information Security South Africa (ISSA) Conference (2006) 4, 6
8. Kuntze, N., Schmidt, A.U.: Employing Trusted Computing for the forward pricing of pseudonyms in reputation systems. To appear in: Proceedings of the Workshop Virtual Goods at the Conference AXMEDIS 2006, Leeds, England, 13.–15. December 2006 8
9. Anderson, C.: The Long Tail. Wired Issue 12.10 - October 2004 http://web.archive.org/web/20041127085645/http://www.wired.com/wired/archive/12.10/tail.html 8
10. Bakos, Y.: The emerging role of electronic marketplaces on the internet. Commun. ACM **41** (1998) 35–42 8
11. Resnick, P., Kuwabara, K., Zeckhauser, R., Friedman, E.: Reputation systems. Communications of the ACM **43** (2000) 45 – 48 8
12. Axelrod, R.: The Evolution of Cooperation. Basic Books, New York (1984) 8
13. Salganik, M.J., Dodds, P.S., Watts, D.J.: Experimental study of inequality and unpredictability in an artificial cultural market. Science **311** (2006) 854–856 8
14. Dellarocas, C.: Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In: ACM Conference on Electronic Commerce. (2000) 150–157 8
15. Douceur, J.R.: The sybil attack. In Druschel, P., Kaashoek, F., Rowstron, A., eds.: Peer-to-Peer Systems: First InternationalWorkshop, IPTPS 2002 Cambridge, MA, USA, March 7-8, 2002. Volume 2429 of Lecture Notes in Computer Science, Springer-Verlag (2002) 251–260 9
16. Friedman, E.J., Resnick, P.: The social cost of cheap pseudonyms. Journal of Economics & Management Strategy **10** (2001) 173–199 9

17. Dellarocas, C.: Sanctioning reputation mechanisms in online trading environments with moral hazard. MIT Sloan Working Paper No. 4297-03 (2004) 9
18. Bhattacharjee, R., Goel, A.: Avoiding ballot stuffing in ebay-like reputation systems. In: P2PECON '05: Proceeding of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems, New York, NY, USA, ACM Press (2005) 133–137 9
19. Cheng, A., Friedman, E.: Sybilproof reputation mechanisms. In: P2PECON '05: Proceeding of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems, ACM Press (2005) 128–132 9
20. Buttyan, L., Hubaux, J.P.: Accountable anonymous access to services in mobile communication systems. In: Proceedings of the 18th IEEE Symposium on Reliable Distributed Systems. (1999) 384–389 9
21. Zieglera, G., Farkas, C., Lõrincz, A.: A framework for anonymous but accountable self-organizing communities. Information and Software Technology **48** (2006) 726–744 9
22. Trusted Computing Group: Mobile Phone Working Group Use Case Scenarios - v 2.7. Technical report, TCG (2005) 11, 12
23. Open Mobile Alliance: Push architecture. draft version 2.2 - 20 jan 2006. oma-ad-push-v2_2-20060120-d. Technical report, Open Mobile Alliance (2006) 10