Tagging the Turtle: Local Attestation for Kiosk Computing

Ronald Toegl*

Institute for Applied Information Processing and Communications (IAIK)
Graz University of Technology, Austria
ronald.toegl@iaik.tugraz.at

Abstract. Public kiosk computers are especially exposed and the software running on them usually cannot be assumed to be unaltered and secure. The Trusted Platform Module (TPM) as a root of trust in an otherwise untrusted computer allows a machine to report the integrity and the configuration of a platform to a remote host on the Internet. A natural usage scenario is to perform such an *Attestation* prior to handling sensitive or private data on a public terminal.

Two challenges arise. First, the human user needs to reach her trust decision on the basis of the TPM's cryptographic protocols. She cannot trust the public machine to display authentic results. Second, there is currently no way for the user to establish that the particular machine faced actually contains the TPM that performs the Attestation.

In this paper we demonstrate an Attestation token architecture which is based on a commodity smart phone and more efficient and flexible than previous proposals. Further, we propose to add a low-cost Near Field Communication (NFC) compatible autonomic interface to the TPM, providing a direct channel for proof of the TPM's identity and local proximity to the Attestation token.

Keywords: Trusted Computing, Kiosk Computing, Near Field Communication, Attestation.

1 Introduction

Kiosk computers openly available in public places are highly exposed and threatened by a variety of software-based attacks in the form of viruses, key-loggers and root kits. They cannot be trusted to handle sensitive information such as account logins, passwords or other private data.

The Trusted Computing Group (TCG) specified the Trusted Platform Module (TPM), a system component deeply embedded in a machine's hardware and software architecture. One of its mechanisms, called *Remote Attestation*, reports the platform's state to another host. This helps to establish cryptographically qualified and tamper-evident assurance on the software configuration of a machine.

* This work was supported by the European Commission through project OpenTC, FP-6, contract no. 027635 and by the Österreichische Forschungsförderungsgesellschaft through project acTvSM, FIT-IT no. 820848.

J.H. Park et al. (Eds.): ISA 2009, LNCS 5576, pp. 60–69, 2009.

As Attestation allows to determine the absence of malicious software, it is desirable for a user to perform it prior to entering sensitive or confidential data into a public kiosk. However, several challenges need to be overcome to allow Attestation in this scenario. The TPM does not offer a secure local display, so a malicious kiosk might display fake reports on its trustworthiness. McCune et al. [1] propose to equip the user with an ideal, axiomatically trustworthy device, the so-called *iTurtle*, which indicates the security of a device. A practical implementation for the kiosk scenario using smart phones is demonstrated by Garris et al. [2]. Also, the TCG's Attestation scheme does not guarantee that the TPM is located within the machine the user faces. Parno [3] proposes a direct link between the user and the TPM.

In this paper we build on these previous results and introduce two novel improvements. Firstly, considering the resource limitations of mobile devices, the proposed schemes are not flexible and scalable enough. We demonstrate an efficient solution based on smart phones and a trusted third party. Secondly, to include a proof-of-locality in the process, we propose to introduce Near Field Communication (NFC) in the TCG's security architecture.

Outline. The remainder of the paper is organized as follows. In Section 2, we outline Trusted Computing and NFC technologies, discuss related work and present the kiosk scenario we consider. In Section 3 we identify challenges specific to the kiosk scenario and present our Mobile Attestation Token Architecture, discuss integration of NFC in the TPM and present our prototype implementations. The paper concludes in Section 4.

2 Preliminaries

2.1 Trusted Computing

The Trusted Computing Group[1] has specified the Trusted Platform Module. Similar to a smart card, the TPM features cryptographic primitives, but it is physically bound to the main device. A tamper-resilient chip contains hardware primitives for public-key cryptography, key generation, cryptographic hashing, and random-number generation. With these components the TPM is able to enforce its security policies against any remote attacker.

The TPM implements high level functionality such as reporting the current system state and providing evidence of the integrity and authenticity of this measurement, known as *Remote Attestation*. This is done with the help of the Platform Configuration Registers (PCRs), which can only be written to with the *Extend Operation*. A PCR with index i in state t is extended with input x by setting

$$PCR_i^{t+1} = \text{SHA-1}(PCR_i^t || x).$$

Before executable code is invoked, the caller computes the code's hash value and extends a PCR with the result. In this way a *chain of trust* is built, starting from the BIOS, covering bootloader, kernel, system libraries, application code, etc. Ultimately, the exact configuration of the platform is mapped to PCR values. If such a system state

[1] `http://www.trustedcomputinggroup.org`

fulfills the given security or policy requirements, we refer to it as a *trusted state*. To analyze the so collected state information, a Stored Measurement Log (SML) is kept.

When called to perform the *Quote* operation, the TPM signs the current values of the PCRs together with a host-supplied nonce. To protect the platform owner's privacy, the unique Endorsement Key, typically injected by the TPM manufacturer, is not used for this signature. Rather, a pseudonym is used: an *Attestation Identity Key (AIK)*. The authenticity of an AIK can either be certified by a Trusted Third Party (TTP), named PrivacyCA [4], or with the group-signature-based DAA scheme. The AIK certificate vouches that the private signature key is securely held by a standard conforming TPM.

2.2 Near Field Communication

Near Field Communication (NFC) is a recent wireless technology which combines the advantages of passive RFID (Radio Frequency IDentification) systems and active wireless communication. The communication layer is standardized in ISO 18092 [5], ECMA 340 [6] and ECMA 352 [7]. It operates in the 13.56 MHz frequency range, supports data rates at 424kbit/s and uses Amplitude Shift Keying (ASK) modulation. The typical operating distance between two NFC devices is only a few centimeters. In passive mode, the reader device establishes a Radio Frequency (RF) field that is used by the passive participant to send data over the air interface. In active mode, both communication devices generate the field for data transfer. An NFC link is always established between an initiator and a target. In case of active-passive mode, the passive tag always acts as NFC target. NFC is also designed to be compatible with smart-card standards such as ISO 14443 A and B and Sony's FeliCaTM technology. Additionally, NFC technology works together with the ISO 15693 standard for passively powered RFID tags. In contrast to other wireless communication technologies which are designed for a large communication range, NFC enables short-distance communication between electronic devices in a very intuitive way for the user, by simply bringing the devices close together (touching). It follows the very natural principle for communication between only two, locally present entities. The physical security is based on the small operating range of the devices: Communication is only possible in the immediate proximity of the reader with a read/write range of up to 10 cm. A fixed location of an NFC tag (passively or actively powered) can provide evidence whether a mobile NFC device (or its user) has been at that location.

This allows advanced applications including electronic cash, credit-card payment, electronic key, and easy establishment of wireless communication networks like Bluetooth or IEEE 802.11. NFC is supported by most leading manufacturers of mobile consumer electronics (like mobile phones, PDAs, etc.).

2.3 Scenario

We consider kiosk computers, as often found at shops, in hotel lobbies, transportation terminals or Internet cafés. They are public terminals and provide applications like Web browsers or ticket-vending services.

With TPM-based Attestation we desire to provide the user with means to establish trust in a kiosk, but of course we have to consider the limitations of TPM security: it is not designed to protect against hardware attacks. We assume that the devices are unattended, but physically protected, i.e. by robust casings fixed to the ground, integrated keyboards and displays that prevent hardware based attacks. Also, we assume that the operator performs hardware and software maintenance on a regular basis, thus making most hardware attack schemes like adding devices to the casing impractical.

However, also attackers can visit the kiosk repeated times during operation hours and pretend to be legitimate users. Attackers are assumed to have full control over the software running on the kiosk, thus software cannot be trusted at all and keyloggers and fake security tools must be assumed. We further assume that wireless communications can be eavesdropped.

2.4 Related Work

To enable Remote Attestation [8], the TCG standards [9] describe a set of compact basic hardware building blocks that are designed to enable a host to measure the exact software binaries running on it and to report this result to another system on the internet. Operating systems that measure at least a partial chain-of-trust with the help of virtualization have been demonstrated under laboratory conditions, e.g. [10]. Using the quote result, a verifier is expected to come to a trust decision based on a measurement log that holds a list of binaries' file names and their hash code. However, the number of possibly combinations of secure software configurations in today's open system architectures is often too large in practice [11]. Alternative concepts like *Property-based Attestation* [12,13,14] delegate the state analysis to a *Trusted Third Party* (TTP) which issues certificates for specific properties.

Another challenge of Attestation is to report the result to the user, even if the display the user faces cannot be trusted. [3] analyzes the *Cuckoo-attack* (also known as Mafia fraud attack or chess grandmaster problem) where malware on a compromised local machine relays TPM messages to another TPM on a remote machine which is in a trusted state. The author concludes that a local binding between user and TPM is needed. If the user is in possession of a trusted hand-held device, this can be achieved physically via a special hard-wired interface or cryptographically by providing users with a key by means of a sticker on the machine casing. Note, if the remote machine is under physical control of the attacker, a simple hardware TPM reset attack [15] would allow feeding fake measurements to the TPM, circumventing the chain of trust altogether.

[1] propose the concept of an *iTurtle* device which can be trusted axiomatically. To achieve *user-observable verification* it should be as simple as possible, even without support for cryptography and thus easy to understand and certify. Envisioned is a USB device with LEDs indicating the trust status. The authors argue that integration in the TCG's cryptographic scheme is too complex and that the challenges of state analysis on a restricted device remain.

More powerful PDAs and smart phones have demonstrated [16,17] their applicability as trusted portable device to work in conjunction with a trusted server and an untrusted public terminal to act as a secure keyboard and GUI to the user. [18] demonstrate a mobile phone application which uses 2-D barcodes on stickers to identify a public key of devices like printers or IEEE 802.11 access points and also consider integration of

TPM-based Attestation. This architecture does not guarantee the identity and standard-conformance of the TPM, i.e. lacks the validation of AIK certificates.

The specific case of attesting a public kiosk computer using a mobile phone has been studied in detail by [2]. A user wishing to use a kiosk first uses the camera of her smartphone to scan the barcode containing the hash of the AIK certificate of the kiosk. The phone then connects to the kiosk using Bluetooth. The kiosk now transmits a set of configurations it supports. The set is pre-defined and signed by the kiosk's operator, which has to trusted. Now the user chooses a configuration and the kiosk reboots to build a fresh chain-of-trust. After it is online again, the phone performs an Attestation protocol, compares the reported configuration against the chosen one and validates that the `TPM_Quote` result is indeed signed with the same AIK. The user is informed of the result, i.e. the trust status is displayed on her phone. She can then use the kiosk's applications or even supply a private virtual machine image containing her private software and data, cf. [19]. In the end, the user logs out.

3 Local Attestation for Kiosks

Based on the presented literature we identify the following additional challenges for the Attestation of kiosk computers, which occur due to the locality of the attestant.

- **Flexible and Scalable Trust decisions.** The display of a public computer must not be trusted to securely show the trust decision. To convey the trust status to a user, a mobile Attestation token is needed that provides a suitable display and a secure communication channel to the TPM. Existing implementations not only display the result, but also perform the trust decision on the mobile Attestation device. The performance of these devices limits the size of the known-good-value repository and the complexity of the state analysis needed for the trust decision. Also, if only a small set of possible configurations is provided, none of them might match the specific security requirements of the user. A priori stored reference values also limit the flexibility in case of system updates, or when encountering terminals from unexpected operators.
- **Direct, Local Channel between User Token and TPM.** Practical proposals have so far considered different interfaces such as Bluetooth or USB. However, as outlined in [3], both technologies require an honest software stack to forward their messages to the passive TPM device. A direct wired physical channel would require extensive changes to the TPM design and new standard plugs, both being expensive and impractical. Bluetooth has a long radio range and thus it could also connect to a neighboring kiosk. To prevent this, current proposals introduce stickers that identify TPM keys. However, stickers are easy to manipulate [20]. Foremost, it is extremely easy to copy them (with the attacker posing as a legitimate user, taking the photo with his mobile phone camera) and thus fake the identity of another kiosk. This is exactly the setting for the cuckoo attack we need to prevent.

We now present two novel improvements which constitute our main contributions. In the next section we outline a kiosk Attestation architecture which is designed to be user-controlled, flexible and scalable with regard to kiosk state analysis. In Section 3.2 we detail how NFC can be integrated in Attestation, thus providing for direct user token to TPM communications.

3.1 Mobile Attestation Token

In our scheme, three parties collaborate to perform a cryptographic protocol. The *Kiosk* contains a TPM and an operating system that offers a complete chain-of-trust and measurement services that allow the extraction of properties. Secondly, the *Mobile Attestation Token* (MAT) is the client the user installs on his mobile phone. Finally, we introduce a trusted third party, a *Verification Server* (VS).

The protocol flow is shown in detail in figure 1. The user initiates Attestation with her MAT, providing the URL of the Verifications Server she intends to use and a nonce to provide freshness. The kiosk will then gather a Quote from the TPM and forward the result over a secure SSL connection to the VS. It analyzes the quote and stored measurement log of the kiosk and decides the trustworthiness according to the detailed requirements of the user, using its local or other available Known-Good-Value services or property extraction modules. The result is returned as a ticket. It contains a binary trust decisions (the `trusted-bit`) and a free `text` for additional messages. The ticket is validated on the MAT and the final result displayed.

In our architecture there are no limitations on who operates this verification server. It could be the kiosk operator as well as the user or any commercial or open institution. This provides flexible adaption to changing profiles. Also we require the MAT only to validate the signature of the ticket, the nonce and the trusted bit. Thus, state analysis is not limited by the restricted resources on the mobile device.

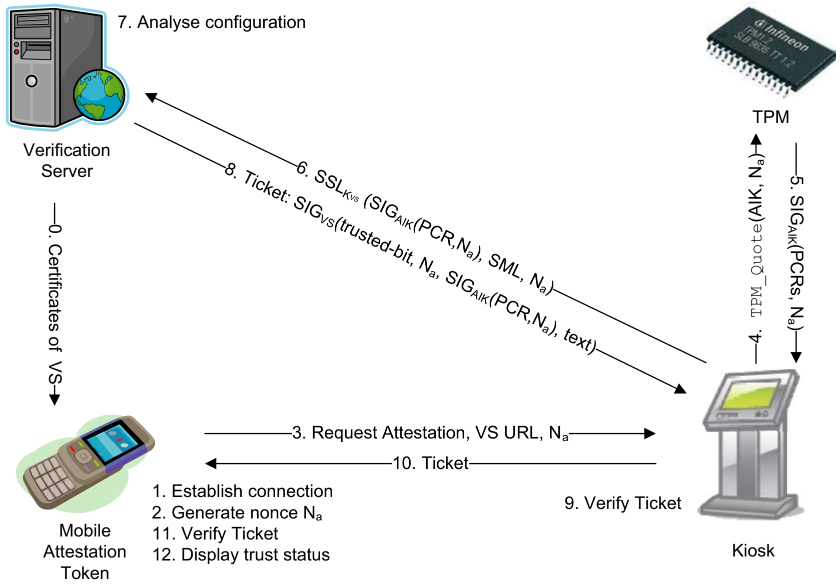


Fig. 1. Ticket-based Local Attestation scheme with MAT and trusted Verification Server

Table 1. The TPM_establishNonce_NFC command establishes a shared nonce between remote NFC reader and TPM. The resulting nonce is not returned to the host machine but retained in the protection of the TPM.

Incoming

Parameter	Size	Type	Name	Description
1	2	TPM_TAG	tag	TPM_TAG_RQU_COMMAND
2	4	UINT32	paramSize	Total number of input bytes including paramSize and tag
3	4	TPM_COMMAND_CODE	ordinal	Command ordinal: TPM_ORD_establishNonce_NFC

Outgoing

Parameter	Size	Type	Name	Description
1	2	TPM_TAG	tag	TPM_TAG_RSP_COMMAND
2	4	UINT32	paramSize	Total number of input bytes including paramSize and tag
3	4	TPM_RESULT	return code	The return code of the operation
4	4	TPM_COMMAND_CODE	ordinal	Command ordinal: TPM_ORD_establishNonce_NFC

3.2 An NFC Interface for the TPM

In this section, we outline how the TPM could be extended with an NFC interface to create a direct channel to the MAT. We believe that this will not require extensive changes to the TPM design. NFC has been designed to be integrated in small hardware solutions like smart cards, which are very similar to many TPM implementations. Furthermore, many of the challenges that have to be overcome in the design of passive NFC tags are not an issue with the TPM. For instance, the TPM has an active power supply and full cryptographic capabilities. Only a simple, passive RF-Interface is needed, and the antenna circuit could just be printed on the mainboard of the host machine[2]. We believed, that it is cheaper than a proprietary wired interface, which would require modifications to TPM, board, casing and MAT.

In this way it is possible to establish a direct link from the Attestation device to the TPM. Note that in our approach any software on the kiosk is circumvented, making software-based attacks on the connection impossible.

New TPM Commands. Also changes to the TPM itself can be limited to a minimum. As a special purpose trusted component, it should not provide more features than necessary to perform its tasks and therefore should not operate as a full flexible NFC reader to the host. Also, changes to the TPM API should be minimal and not affect normal operations. For brevity, we only present the changes to the current TCG TPM specification [9] in this section.

The RF interface is to be activated only in Enabled-Activated-Owned state of the TPM lifecycle and an owner-authorized call to TPM_SetCapability is needed to activate the permanent flag enableNFCInterface that enables the following operations.

We introduce a new command that allows the NFC reader and the TPM, which have no prior knowledge of each other's identity, nor a shared key, to jointly establish a shared secret over the NFC channel. This secret can then be used as nonce in a *single* subsequent TPM operation. The TPM_establishNonce_NFC command is described in Table 1. It is important to notice that if the command returns with TPM_SUCCESS, the nonce is not returned to the TPM's host machine but retained in a special volatile

[2] Assuming a non-shielded casing.

and protected register TPM_NFC_NONCE inside the TPM. This register can be read-accessed as if it was an additional PCR, but with one exception: it is always reset to zero after a read operation. If the protocol fails, or times out, appropriate error codes are returned. TPM_establishNonce_NFC does not require authorization, as it only stores the nonce. All commands, that use its result must be properly authenticated. The command itself performs a standard Diffie-Hellman key exchange.

Minor changes are now needed for TPM commands that utilize this nonce, for instance, TPM_Quote. It is called with a TPM_PCR_SELECTION that indicates the PCR registers to consider. The behavior is extended as follows: TPM_NFC_NONCE is selected like other PCRs with index: number of normal PCRs + 1. If TPM_NFC_NONCE is zero, the command terminates with error code TPM_NO_NFC_NONCE, else its value is hashed together with the PCRs and signed with the provided AIK. The values of all used registers are returned to the host. The TPM_NFC_NONCE register is then set to zero.

This way, the quote result depends on the TPM_NFC_NONCE that was previously agreed upon by the NFC reader, i.e. the Mobile Attestation Token and the TPM. As the Quote result is signed with an AIK, this links TPM_NFC_NONCE to an authentic TPM. Each nonce can only be used once, thus guaranteeing freshness. Other commands which access PCRs can be adapted in a similar way, without changing their signature.

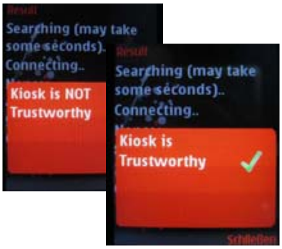
NFC Security Considerations. The security of NFC is mainly based on the physical characteristics of the electromagnetic near field, which limit the operational range to about 10 cm. It should be noted, especially as the physical layer is typically not encrypted, that eavesdropping might be possible, even at a distance. Still, classical Man-in-the-Middle Attacks can be prevented in active-passive mode due to the characteristics of transmission parameters, which allow the reader to sense manipulations in field [21].

However, on RFID-systems closely similar to NFC, relay attacks have been demonstrated [22] using custom built hardware. Since then a number of proposals have presented various Distance Bounding Protocols [23,24,25] which allow to prevent relay attacks. Commonly, they utilize a pre-shared key K_{DBP} and play a challenge-response protocol based on it to measure the message runtime. K_{DBP} could be easily and securely distributed [26] to the parties in our Attestation scheme. Yet, due to the small distance in NFC the distance bounding protocols are a challenge to implement, with proposals requiring different modulation schemes, UWB channels and high-precision clocks. Thus, distance bounding implementations are more expensive than standard NFC and not generally available yet.

3.3 Implementation Details

We base our prototype implementation on commodity hardware and on platform-independent software. On the Kiosk and on the Verification Server, Java SE is used. The verification server stores reference Known-Good-Values in a rational MySQL database, which is accessed using Hibernate. We supply a GUI tool to allow the user to collect reference measurements. On the Kiosk, we currently collect binary measurements, and accept plug-ins for trust property analysis. The TPM can be accessed using IAIK jTSS[3].

[3] http://trustedjava.sf.net/



Result

Searching (may take some seconds)..

Connecting..

Kiosk is NOT Trustworthy

Result

Searching (may take some seconds)..

Connecting..

Kiosk is Trustworthy ✓

Fig. 2. The Mobile Attestation Token software informs the user on the result of the Attestation process in a comprehensible way

The Mobile Attestation Token is built as applet for Java ME, MIDP 2.0, extended with JSR 82 (Bluetooth/OBEX support), JSR 75 (PDA profile) and JSR 257 (NFC support). For cryptographic support we use IAIK JCE ME on all hosts. The MAT software is thus compatible to NFC-enabled phones such as the Nokia 6212. Figure 2 shows a typical screenshot on the MAT. As no NFC-enabled TPM is currently available in hardware, we simulate all communications using Bluetooth only.

4 Conclusions

In this paper we consider challenges that arise to the Attestation of public kiosk computers. We extend on previous proposals to provide more scalability considering the limited computational power and memory of mobile devices, and improve flexibility by moving the complex kiosk state analysis to a trusted third party. While it does not overcome all complexities of Attestation, our scheme allows for full user control over security requirements and trust policies.

Furthermore, we propose to add a direct, affordable interface to the TPM. With Near Field Communication, a proof of locality can be included in the Attestation process. We also discuss the security implications of using NFC. The presented extension allows us to completely circumvent any malicious software and thus prevent ”Cuckoo” attacks on kiosk computers.

Acknowledgements. The author thanks Manuel Schallar and Herwig Guggi for assisting with the implementation of the MAT, Daniel M. Hein for valuable comments and Michael Hutter and Martin Feldhofer for insightful discussions on NFC.

References

1. McCune, J.M., Perrig, A., Seshadri, A., van Doorn, L.: Turtles all the way down: Research challenges in user-based attestation. In: Proceedings of HotSec. USENIX Association (2007)
2. Garriss, S., Cáceres, R., Berger, S., Sailer, R., van Doorn, L., Zhang, X.: Trustworthy and personalized computing on public kiosks. In: MobiSys, pp. 199–210. ACM Press, New York (2008)
3. Parno, B.: Bootstrapping trust in a ”trusted” platform. In: Proc. of HotSec. USENIX (2008)

4. Pirker, M., Toegl, R., Hein, D., Danner, P.: A PrivacyCA for anonymity and trust. In: Chen, L., Mitchell, C.J., Martin, A. (eds.) Trust 2009. LNCS, vol. 5471, pp. 101–119. Springer, Heidelberg (2009)
5. Iso/iec 18092:2004 – near field communication – interface and protocol (nfcip-1). International Organization for Standardization (2007)
6. ECMA: ECMA-340: Near Field Communication — Interface and Protocol (NFCIP-1). European Association for Standardizing Information and Communication Systems (2004)
7. ECMA: ECMA-352: Near Field Communication Interface and Protocol-2 (NFCIP-2). European Association for Standardizing Information and Communication Systems (2003)
8. Coker, G., Guttman, J., Loscocco, P., Sheehy, J., Sniffen, B.: Attestation: Evidence and trust. In: Chen, L., Ryan, M.D., Wang, G. (eds.) ICICS 2008. LNCS, vol. 5308. Springer, Heidelberg (2008)
9. Trusted Computing Group: TCG TPM specification version 1.2 revision 103 (2007)
10. Sailer, R., Zhang, X., Jaeger, T., van Doorn, L.: Design and implementation of a tcg-based integrity measurement architecture. In: Proc. of Security 2004. USENIX (2004)
11. England, P.: Practical techniques for operating system attestation. In: Lipp, P., Sadeghi, A.-R., Koch, K.-M. (eds.) Trust 2008. LNCS, vol. 4968, pp. 1–13. Springer, Heidelberg (2008)
12. Sadeghi, A.R., Stüble, C.: Property-based attestation for computing platforms: caring about properties, not mechanisms. In: Hempelmann, C., Raskin, V. (eds.) NSPW. ACM Press, New York (2004)
13. Chen, L., Landfermann, R., Löhr, H., Rohe, M., Sadeghi, A.R., Stüble, C.: A protocol for property-based attestation. In: Proccedings of STC. ACM Press, New York (2006)
14. Kühn, U., Selhorst, M., Stüble, C.: Realizing property-based attestation and sealing with commonly available hard- and software. In: Proccedings of STC. ACM Press, New York (2007)
15. Kauer, B.: Oslo: improving the security of trusted computing. In: Proceedings of 16th USENIX Security Symposium, pp. 1–9. USENIX Association (2007)
16. Oprea, A., Balfanz, D., Durfee, G., Smetters, D.K.: Securing a remote terminal application with a mobile trusted device. In: Yew, P.-C., Xue, J. (eds.) ACSAC 2004. LNCS, vol. 3189. Springer, Heidelberg (2004)
17. Sharp, R., Scott, J., Beresford, A.: Secure mobile computing via public terminals (2006)
18. McCune, J., Perrig, A., Reiter, M.: Seeing-is-believing: using camera phones for human-verifiable authentication. In: 2005 IEEE Symposium on Security and Privacy (2005)
19. Cáceres, R., Carter, C., Narayanaswami, C., Raghunath, M.: Reincarnating PCs with portable soulpads. In: Proc. of MobiSys, pp. 65–78. ACM Press, New York (2005)
20. Lindner, F.: Toying with barcodes. In: 24th Chaos Communication Congress (2007)
21. Haselsteiner, E., Breitfuss, K.: Security in near field communication (nfc). In: Workshop on RFID Security (2006)
22. Hancke, G.: A practical relay attack on iso 14443 proximity cards. Technical report, University of Cambridge (2005)
23. Tu, Y.J., Piramuthu, S.: Rfid distance bounding protocols. In: First International EURASIP Workshop on RFID Technology (2007)
24. Reid, J., Nieto, J.M.G., Tang, T., Senadji, B.: Detecting relay attacks with timing-based protocols. In: Proceedings of ASIACCS 2007, Singapore, pp. 204–213. ACM Press, New York (2007)
25. Munilla, J., Peinado, A.: Distance bounding protocols for RFID enhanced by using void-challenges and analysis in noisy channels. In: Wirel. Commmun. Mob. Comput. 2008, vol. 8, pp. 1227–1232. Wiley Interscience, Hoboken (2008)
26. Toegl, R., Leung, A., Hofferek, G., Greimel, K., Phan, R., Bloem, R.: Formal analysis of a TPM-based secrets distribution and storage scheme. In: Proceedings of TrustCom 2008. IEEE Computer Society Press, Los Alamitos (2008)