# A Trusted Platform Module for Near Field Communication

Michael Hutter and Ronald Toegl
Institute for Applied Information Processing and Communications (IAIK)
Graz University of Technology, Inffeldgasse 16a, 8010 Graz, Austria
Email:{Michael.Hutter,Ronald.Toegl}@iaik.tugraz.at

***Abstract*—Near Field Communication (NFC) has become widely available on smart phones. It helps users to intuitively establish communication between local devices. Accessing devices such as public terminals raises several security concerns in terms of confidentiality and trust. To overcome this issue, NFC can be used to leverage the trusted-computing protocol of remote attestation.**

**In this paper, we propose an NFC-enabled Trusted Platform Module (TPM) architecture that allows users to verify the security status of public terminals. For this, we introduce an autonomic and low-cost NFC-compatible interface to the TPM to create a direct trusted channel. Users can access the TPM with NFC-enabled devices. The architecture is based on elliptic-curve cryptography and provides efficient signing and verifying of the security-status report. As a proof-of-concept, we implemented an NFC-enabled TPM platform and show that a trust decision can be realized with commodity smart phones. The NFC-enabled TPM can effectively help to overcome confidentiality issues in common public-terminal applications.**

***Keywords*-Trusted Computing; RFID Security; Near Field Communication; NFC; ECDSA; Remote Attestation**

## I. Introduction

During the last decade, mobile technology has grown significantly offering many applications including payment, ticketing, and access control. Especially Near-Field Communication (NFC) has become widely available on the market offering convenient services by simply touching NFC-enabled objects in the proximity with commodity mobile phones. This article presents a method to overcome confidentiality and trust issues in security-related NFC applications.

Trusted Computing provides services to establish trust in networked environments. One service, as devised by the Trusted Computing Group (TCG) industry consortium, is *remote attestation*. It achieves trust decisions between hosts. A TPM is therefore used to measure the integrity and confidentiality guaranteeing the status of a targeted host platform. This status must then be reported to the user who decides if the platform can be trusted or not. Unfortunately, the cryptographic protocols that actually perform the attestation do neither provide a human intelligible conveyance of the status report nor an intuitive identification of the targeted host platform. In a mobile user scenario, small and portable devices such as mobile phones or PDAs can perform the attestation protocol reporting the status of the targeted platform to the user [1], [2], [3], [4]. However, to provide this service, a secure channel between the user and the platform-integrated TPM is mandatory as stated by Parno [5].

In this article, we propose to integrate both an NFC interface and a TPM into one hardware component. The integration provides a secure trust decision to the user by guaranteeing the physical presence of the user through NFC and the direct channel to the TPM by the combination of both modules in one piece of silicon. The proposed architecture leverages the remote attestation protocol so that a mobile phone can be used to establish trust to a public terminal in the proximity. We improve on a previously reported proposal [4] by moving the trust decision into the mobile device. This is enabled by using an advanced terminal hardware and software platform and a more efficient cryptographic protocol. Therefore, we propose to use elliptic curve cryptography (ECC) to increase the computation and communication performance. As a proof-of-concept, we implemented an NFC-enabled TPM prototype that performs remote attestation using the elliptic curve digital signature algorithm (ECDSA). We further give performance results of our practical experiments.

The remainder of this paper is structured as follows. In Section II, we introduce Trusted Computing and NFC technologies. Section III describes the proposed architecture. We present implementation details and performance results in Section IV. The paper concludes in Section V.

## II. Trusted Computing in NFC Environments

NFC is a wireless communication technology that provides a platform for many applications such as mobile payment, ticketing, and access control. One key feature of NFC is the simple data acquisition just by touching an object with an NFC-enabled reader. Such readers might be integrated in mobile phones or digital cameras that transfer information to the devices in their proximity. There exist two different modes for a communication between a reader (initiator) and a target device. In passive communication mode, the initiator provides an electromagnetic (EM) field which is used to power the target device and which allows a bidirectional communication. In active communication mode, both the initiator and the target device provide an alternately generated EM field so that both devices require an active power supply.

NFC is based on the Radio Frequency Identification (RFID) technology that operates at 13.56 MHz frequency. As opposed to RFID, NFC follows several specifications that have been

 DOI: http://dx.doi.org/10.1109/ICSNC.2010.27

standardized by the International Organization for Standardization (ISO) and the European Computer Manufacturers Association (ECMA). These standards specify the used data modulation, coding, frame formats, data rates, and also the application-specific transport protocol. The typical operating distance between two NFC devices is only a few centimeters (up to 10 cm). Thus, fixing an NFC device (passively or actively powered) to a fixed location can provide evidence whether a mobile NFC device (or its user) has been at that location or not. Besides this evidence, NFC offers a very intuitive way for the user to communicate with a target object by simply bringing the devices close together (touching). It follows the very natural principle for communication between only two, locally present entities.

When two entities would like to establish a connection using NFC, the questions of integrity and confidentiality rise inevitably. Especially in security-related applications like mobile payment, cryptographic services are needed that provide a decision mechanism to the user if the connected device can be trusted or not. Compromised devices pose a serious thread that might extract secret information, perform unwanted payment transactions or behave untrustworthy in some other way.

Trusted computing offers such trust decisions by integrating a so-called Trusted Platform Module (TPM) [6] into target machines, e.g. public terminals. The Trusted Computing Group (TCG) has specified the TPM for general purpose computer systems. Similar to a smart card, the TPM features cryptographic primitives but it is physically bound to its host device. A tamper-resilient integrated circuit contains implementations for public-key cryptography, key generation, cryptographic hashing, and random-number generation, thus providing a root of trust.

In particular, the TPM implements high-level functionality such as reporting the current system configuration and providing evidence of the integrity and authenticity of this measurement. This service is also known as *Remote Attestation*. During the remote attestation process, the TPM receives hashes of several system-state values and stores the hashes in dedicated Platform Configuration Registers (PCRs) located in the TPM. A PCR with index $i$ in state $t$ is extended with input $x$ by setting

$$PCR_i^{t+1} = \text{SHA1}(PCR_i^t || x).$$

The basic operation of a TPM is as follows. Before executable code is invoked, a hash value of the code is computed and stored in a PCR. Modern platforms from AMD [7] and Intel [8] extend this mechanism, by the option of a *dynamic* switch to a known-secure system state. A so-called *late launch* is initiated by a special CPU instruction. It stops all parallel processing cores and locks all memory. A special Authenticated Code Module (ACM) sets the platform into a well-defined state. Subsequently, a Measured Launch Environment (MLE) [9] may be measured into the PCRs and the control is passed to it.

Normal platform operation can be restored step-by-step with careful release of resources and measurements of code and settings. Ultimately, the exact configuration of the platform is mapped to PCR values. This property makes it impossible to hide a malicious program on a thus protected computer. If such a system state fulfills the given security or policy requirements, we refer to the system state as a *trusted state*.

The TPM is capable of signing the current values of the PCRs together with a supplied nonce. This is called the *Quote* operation. To protect the platform owner's privacy, a pseudonym identity is used: an *Attestation Identity Key (AIK)*. The authenticity of an AIK can be certified by an on-line trusted third party, called PrivacyCA [10], or with the group-signature-based DAA scheme. Then, a remote verifier may analyze the Quote result and decide whether to trust the given configuration or not.

### A. Remote Attestation over NFC

Attestation is useful to improve the security for a number of computing services, including not only remote but also physically present systems. In general, various types of systems may be encountered in different usage scenarios. For instance, a user might want to learn if a public general-purpose desktop computer is secure for ad-hoc use. Customers would like to be assured that a point-of-sales terminal in a shop will not collect their PIN together with the information on the magnetic stripe of their credit card for later frauds. The same holds true for other types of Automatic Teller Machines (ATMs) and payment terminals. Vending machines could be reconfigured by attackers to collect cash but not to release their goods. Other security critical applications may also be found in embedded systems or even peripherals like printers or access points. Here, a service technician might find a method to identify the exact software configuration and its integrity to be useful. In addition, giving voters a method to validate that electronic voting machines have not been tampered might assist to add trust to a poll's outcome.

Public terminals are typically located at shops, in hotel lobbies, transportation terminals, or Internet cafés. They provide different services to users like Internet access via Web browsers or ticket-vending services. These terminals are publicly available and can be accessed by users several times always pretending a legitimate user. Possible attackers are therefore assumed to have full access over the software running on the terminals. As a result, the terminal can not be trusted. The users get no guarantee about the confidentiality and privacy status of the terminal they would like to use.

With TPM-based attestation, trust can be established between users and terminals. There exist several proposals for that. McCune et al. [1] pointed out that it would be desirable to equip the user with a simple, ideal and axiomatically trustworthy device. It would then indicate the security of a device to the user. Molnar et al. [11] describe an RFID-based solution. They proposed to integrate the remote attestation protocol into RFID readers to allow the verification of the privacy status of the reader to existing tags (or users) in the field. Li et al. [12] proposed to secure mobile-payment

applications with remote attestation. They present practical results of an attestation scenario using NFC mobile phones. A similar approach has been presented by Garriss et al. [2]. They designed a protocol by which a mobile phone can determine the integrity of running software on a terminal (kiosk computer).

However, many of the proposed architectures suffer in the fact that the TCG's attestation protocol does not guarantee that the TPM is still located within the machine the user faces. This allows possible machine-in-the-middle attacks where an attacker establishes an indirect link between the user and the TPM over a distrusted channel.

## III. The NFC-Enabled TPM Architecture

In the following section, we briefly outline our design for an NFC-enabled TPM architecture. In contrast to existing publications, we propose to integrate the functionality of a low-cost passive NFC interface into the TPM. This allows users to remotely audit the privacy status of the terminal (target) using a conventional NFC device (initiator) by ensuring a direct channel between the user and the TPM.

In general, passive NFC devices are designed to meet low resource requirements. They consume only a few microwatts of power to provide a certain reading range and they need only a low area so that they can be produced in a large scale with low costs. TPMs, in contrast, have an active power supply and provide many resources that can be even reused by the NFC module. Next to the module, the TPM has to provide an additional RF interface so that an antenna can be simply connected or printed on the main board of the terminal.

The integration of the NFC interface into the TPM additionally gives a guarantee that the targeted terminal is still in the proximity of the user due to the use of NFC. Moreover, it gives a guarantee that the TPM is located in the proximity because the NFC module is physically connected and integrated into the TPM. This avoids machine-in-the-middle attacks. Parno [5] therefore proposed to integrate a special-purpose hardware interface to the TPM to establish a direct link between the user and the TPM so that human inter-actors can themselves establish the proximity of the attesting machine.

Our proposal of integrating an NFC interface into the TPM allows users to verify the integrity of public terminals by easily touching them with an NFC-enabled mobile phone. The trust decision is made using the following trusted computing protocol of remote attestation.

### A. The Remote Attestation Protocol

The first step in the attestation protocol is to generate a nonce $N_0$ with the NFC-enabled mobile phone which acts as a reader device (initiator). As soon as the mobile phone touches the RF antenna of the terminal, the nonce is transmitted over the air interface within a configuration challenge (the nonce serves as fresh data to avoid replay attacks). After that, the public terminal (target) responds with the Quote of its currently recorded terminal state. The Quote, i.e. $Sig_{AIK}(PCR_n..PCR_m, N_0)$, $1 \leq n \leq m \leq 24$, the signature over the selected PCR registers under an Attestation Identity Key $AIK$ of the TPM, is then returned to the mobile phone. The protocol flow is shown in Figure 1. Note that the figure shows only a very compact protocol flow that neither includes issues of key management nor the handling of certificates and trusted third party (TTP) services. These issues have to be addressed in practical implementations to meet all requirements of a comprehensive NFC-attestation solution.

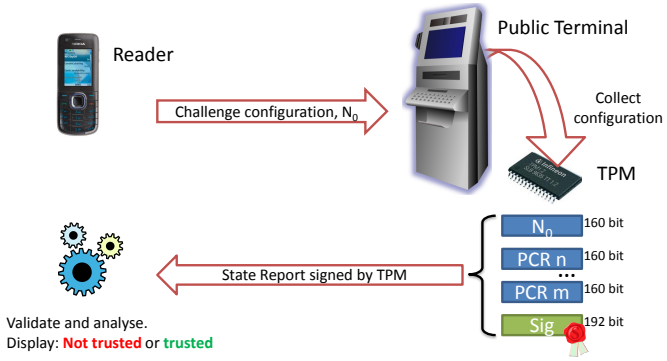


Fig. 1. The compact NFC attestation protocol.

In order to sign the PCRs of the TPM prototype, we propose the use of elliptic curve cryptography (ECC). In contrast to other public-key primitives like RSA, ECC has gathered much attention due to the use of shorter key sizes. The computation time and especially the communication time over the air interface can be significantly improved by providing the same cryptographic strength. For instance, the strength of a 2048 bit RSA key can be compared with a 224 bit ECC key.

Due to that reasons, we propose to use ECDSA to sign the data. The use of ECDSA has several advantages. First, the protocol has been standardized by several organizations such as ANSI, IEEE, NIST, and ISO/IEC. Second, there already exist public-key infrastructures that support that algorithm for signing and verifying data and X.509 certificates.

The algorithm for generating digital signatures is shown in Algorithm 1. It takes a message $m$ as input (containing $N_0$ and the TPM-Quote data structure) and outputs the digital signature $(r, s)$ of that message. The private key $d$ is securely stored in non-volatile memory. The most time consuming operation in ECDSA is the elliptic curve (EC) point multiplication $[k]P$. It takes more that 80 % of the total execution time. Next to that operation, a message digest algorithm (SHA-1) is used to hash the input message. The final signing process needs several finite-field operations such as modular addition, modular multiplication, and modular inversion.

## IV. Implementation Results

In the following, we present results of an implementation of the proposed system architecture. First, we describe the implementation of a public terminal that runs on an attestation-friendly software platform. Second, we describe an NFC-enabled TPM prototype that can be touched by NFC-enabled

**Algorithm 1** Signature-generation scheme using ECDSA

**Require:** Domain parameters, private key $d$, message $m$.
**Ensure:** Signature $(r, s)$

1: Select $k \in [1, n-1]$
2: Compute $[k]P = (x_1, y_1)$, convert $x_1$ to an integer $\bar{x}_1$
3: Compute $r = \bar{x}_1 \mod n$. If $r = 0$ then go back to 1.
4: Compute $e = SHA1(m)$.
5: Compute $s = k^{-1}(e + dr) \pmod n$. If $s = 0$ then go back to step 1.
6: Return $(r, s)$

mobile phones. Third, we show an attestation scenario example and give results about the implementation performance.

### A. The Public Terminal

As a public terminal, we used an attestation-friendly hardware platform that is based on the Intel Trusted Execution Technology (TXT). We measure and enforce the terminal integrity with the acTvSM software platform [13], [14]. It relies on a TPM to provide basic Trusted Computing services such as secure storage of software measurements and on hardware-based virtualization to execute programs in a trusted environment. Right at system boot, a so-called *late launch* is performed using the Intel Trusted Boot open source implementation [15]. We use IAIK jTpmTools and IAIK jTSS [16] tools for configuration. Next, a low-footprint Linux operating system is started from a measured file image using the Measured Launched Environment (MLE). The operating system runs the KVM/Qemu hypervisor [17] which is capable of protecting guest compartments using hardware isolation. The terminal application is, together with its operating system, contained in another image file, which is also measured before execution. From within the virtual machine, we can access the TPM using jTSS to retrieve the Quote that reflects the system state. Note that it is composed of only a few hashes from well-known (read-only) software images and can therefore be compared easily with reference values in NFC-enabled mobile phones.

### B. The NFC-Enabled TPM Prototype

In order to demonstrate an autonomic and NFC-compatible TPM that runs on the public terminal, we developed a low-cost NFC prototype. The prototype simply represents a TPM that is assembled on a Printed Circuit Board (PCB). In contrast to conventional TPM modules, which are sealed and protected against modifications, we used an 8-bit microcontroller for our demonstration that can be freely programmed over a standard JTAG interface. The microcontroller has 128 kB of Flash memory, 4 kB of RAM, and operates at 13.56 MHz. Furthermore, the microcontroller is directly connected to an analog antenna circuit that has been also integrated on the PCB. It has been designed according to ISO/IEC 7810 and has a size of a conventional smart card (ID-1 format). This interface provides an easy access point for NFC-enabled devices. For debugging purposes, there also exists a serial interface on the PCB that allows a communication between the TPM prototype and a PC. In Figure 2, a picture of our NFC prototype is shown where it gets touched by an NFC-enabled mobile phone.

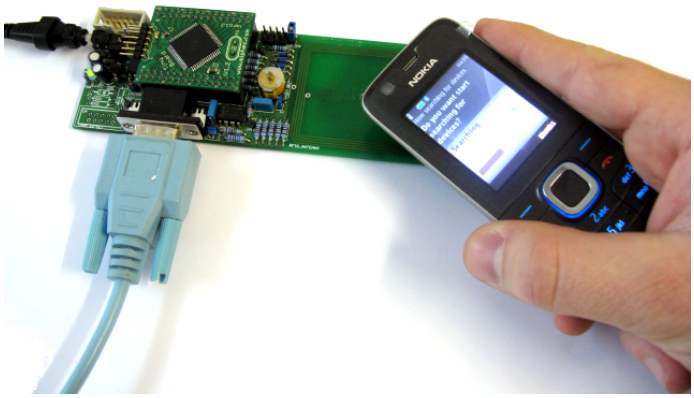


Fig. 2. The NFC-enabled TPM prototype.

For RFID and NFC communication, our TPM prototype implements several protocol standards. It implements RFID protocols such as ISO/IEC 15693, ISO/IEC 14443 (type A and B), ISO/IEC 14443-4 and also ISO/IEC 18092. The software is written in C while parts have been implemented in Assembly language due to timing constraints. Moreover, it implements a user-command interface that allows easy administration over the serial interface. For our experiments, we have used the ISO/IEC 14443-A [18] protocol standard up to layer 4 using ISO/IEC 7816-4 Application Protocol Data Units (APDU) according to the NFC Forum type 4 tag operation specification [19].

In order to sign the PCRs of the TPM prototype, we implemented ECDSA according to the recommendations of the National Institute of Standards and Technology (NIST). The implementation is based on the digital-signature standard [20] and uses a 192-bit elliptic curve over prime fields. As a point multiplication method, we implemented the improved Montgomery ladder proposed by Izu, Möller, and Takagi [21]. The implementation combines the needed point addition and point doubling formulas to one operation needing four intermediate variables to store two curve points $(X_d, Z_d, X_{d+1}, Z_{d+1})$. Note that no Y-coordinates have to be maintained during point multiplication. All finite-field operations have been implemented in C except the finite-field modular multiplication algorithm which was implemented in Assembly language due to performance reasons. The multiplication algorithm uses a product scanning form (Comba multiplication) and applies the fast NIST reduction algorithm to reduce the result. For modular inversion, we applied the Barrett reduction method.

### C. The NFC Attestation Scenario

In the NFC attestation scenario, an NFC-enabled device is used to touch the antenna of the TPM prototype. For this scenario, we used the NFC edition of the Nokia 6212 mobile phone. It is shipped with an integrated RFID-reader chip that allows touching of NFC-enabled objects in the near proximity. Using the Nokia Software Developer Kid (SDK), we implemented a Java J2ME Midlet that runs on the phone. We implemented three threads: `SearchThread`, `SignatureGeneratingThread`, and `SignatureVerifyingThread`. The `SearchThread` handles the detection of passive NFC devices in the field. If our NFC prototype gets touched by the phone, it is detected by the `DiscoveryManager` and an `ISO14443Connection` is established by the Midlet. After that, the Midlet sends an ISO 7816-4 APDU to the prototype to start the attestation process. The prototype signs the PCR values together with the nonce $N_0$. For this, we implemented the same algorithm of ECDSA in a Java Midlet allowing signing and also verifying of digital signatures. The used ECC parameters such as the type of elliptic curve, the curve parameters ($a$ and $b$), or the base point $P$ have been fixed for both devices. After signing, the NFC-enabled TPM responds with the generated quote, i.e. the Quote PCR values and their digital signature $Sig_{AIK}$. The mobile phone compares the received PCR values with reference values and verifies the signature using the public key of the AIK. Note that the phone also verifies the public-key certificate of the AIK that was signed by a PrivacyCA. This certificate can be transmitted over the air interface or can be installed together with the application Midlet that is used to perform the attestation with public terminals. A screenshot of the mobile phone application is shown in Figure 3.

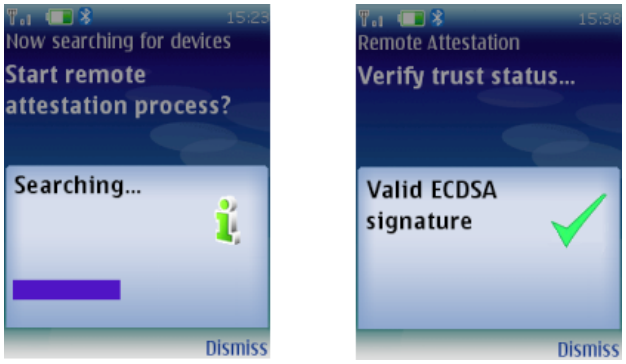


Fig. 3. Screenshots of the remote attestation procedure.

### D. Performance

The digital-signature generation of our TPM prototype takes about 34 million clock cycles. Due to the characteristics of our scenario it is sufficient to consider only the performance of a single session. Running at 13.56 MHz this results in a running time of about 2.57 seconds to generate the signature. The verification of the signed message on the mobile phone takes 33 ms. The transmission time over the air interface has been measured using an 8-bit digital oscilloscope from LeCroy. The time between the first bit transmitted and the last bit received has been taken. First, we measured the anti-collision and initialization phase of the NFC protocol which needs about 22 ms in our experiments. Second, the challenge $N_0$ and the Quote response are transmitted. For this, we assumed a typical number of different PCR values, i.e. 6 in our experiments, resulting in 160+6*160+192=1312 bits. The transmission takes about 140 ms (using a fixed RFID data rate of 106 kbit/s). Thus, the entire attestation process can be performed within 3 second. Note that we focused on a proof-of-concept realization that provides a practical demonstration of the proposed system architecture. Instead of a hardware implementation of the protocol, we implemented all routines in software. Existing TPMs already include cryptographic services such as RSA, where much more bits would have to be transmitted in contrast to elliptic curve implementations. As a comparison, the transmission of a 1024-bit RSA-based signature (comparable with a 160-bit ECC implementation) would need 2192 bits and roughly 240 ms transmission time which is almost twice as high as compared to the elliptic-curve based attestation protocol. This motivated our design decision, as we desire to keep the time the user needs to touch the public terminal as short as possible.

## V. Conclusion

In this article, we proposed an NFC-enabled TPM architecture that allows users to verify the configuration state of public terminals. A trust decision is made and displayed to the user by applying the trusted computing primitive of remote attestation. We introduced a low-cost NFC-compatible interface to the TPM that signs the status report using ECDSA. Furthermore, we implemented a proof-of-concept NFC prototype that shows the practical realizability of our architecture and gives performance results for a trust decision using an NFC-enabled mobile phone. It shows that the use of trusted computing in NFC environments can effectively help to overcome confidentiality issues before the establishment of a potential distrusted terminal session.

## Acknowledgments

The authors would like to thank Markus Pelnar for his valuable support in the practical realization of our investigations. The work has been supported by the European Commission funded project *Collaboration at Rural*, grant no. 034921, and by the Austrian government FIT-IT funded project *acTvSM*, grant no. 820848.

## References

[1] J. M. McCune, A. Perrig, A. Seshadri, and L. van Doorn, "Turtles all the way down: Research challenges in user-based attestation," in *Proceedings of the Workshop on Hot Topics in Security (HotSec)*, August 2007. [Online]. Available: http://www.truststc.org/pubs/286.html

[2] S. Garriss, R. Cáceres, S. Berger, R. Sailer, L. van Doorn, and X. Zhang, "Trustworthy and personalized computing on public kiosks," in *MobiSys '08: Proceeding of the 6th international conference on Mobile systems, applications, and services*. New York, NY, USA: ACM, 2008, pp. 199–210.

[3] R. Toegl, "Tagging the turtle: Local attestation for kiosk computing," in *Advances in Information Security and Assurance*, ser. Lecture Notes in Computer Science, J. H. Park, H.-H. Chen, M. Atiquzzaman, C. Lee, T. hoon Kim, and S.-S. Yeo, Eds., vol. 5576. Springer Berlin / Heidelberg, 2009, pp. 60–69.

[4] R. Toegl and M. Hutter, "An approach to introducing locality in remote attestation using near field communications," *The Journal of Supercomputing*, pp. –, 2010, in print. [Online]. Available: http://dx.doi.org/10.1007/s11227-010-0407-1

[5] B. Parno, "Bootstrapping trust in a "trusted" platform," in *Proceedings of the 3rd conference on Hot topics in security*. San Jose, CA: USENIX Association, 2008, pp. 1–6.

[6] Trusted Computing Group, "TCG TPM specification version 1.2 revision 103," 2007.

[7] Advanced Micro Devices, *AMD64 Virtualization: Secure Virtual Machine Architecture Reference Manual*, May 2005.

[8] D. Grawrock, *Dynamics of a Trusted Platform: A Building Block Approach*, D. J. Clark, Ed. Intel Press, Intel Corporation, 2111 NE 25th Avenue, JF3-330, Hillsboro, OR 97124-5961: Richard Bowles, February 2009, no. ISBN 978-1934053171.

[9] Intel Corporation, "Intel Trusted Execution Technology Software Development Guide," December 2009. [Online]. Available: http://download.intel.com/technology/security/downloads/315168.pdf

[10] M. Pirker, R. Toegl, D. Hein, and P. Danner, "A PrivacyCA for anonymity and trust," in *Trust '09: Proceedings of the 2nd International Conference on Trusted Computing*, ser. LNCS, L. Chen, C. J. Mitchell, and M. Andrew, Eds., vol. 5471. Springer Berlin / Heidelberg, 2009.

[11] D. Molnar, A. Soppera, and D. Wagner, "Privacy for RFID Through Trusted Computing," in *ACM Workshop On Privacy In The Electronic Society, WPES, Alexandria, Virginia, USA, November, 2005, Proceedings*. ACM Press, November 2005, pp. 31–34. [Online]. Available: http://www.cs.berkeley.edu/~dmolnar/papers/wpes05-camera.pdf

[12] Q. Li, X. Zhang, J.-P. Seifert, and H. Zhong, "Secure Mobile Payment via Trusted Computing," in *Asia-Pacific Trusted Infrastructure Technologies – APTC, Third International Conference, October 14 - 17, 2008, Wuhan, China, Proceedings*. Hubei: IEEE, November 2008, pp. 98–112. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4683087&tag=1

[13] M. Pirker and R. Toegl, "Towards a virtual trusted platform," *Journal of Universal Computer Science*, vol. 16, no. 4, pp. 531–542, 2010, http://www.jucs.org/jucs_16_4/towards_a_virtual_trusted.

[14] M. Pirker and R. Toegl, "Dynamic enforcement of platform integrity (a short paper)," in *Proc. 3rd International Conference on Trust and Trustworthy Computing (TRUST 2010)*, ser. LNCS, vol. 6101. Springer Verlag, 2010, (in print).

[15] Intel Corporation, "Trusted Boot," 2008. [Online]. Available: http://sourceforge.net/projects/tboot/

[16] M. Pirker, R. Toegl, T. Winkler, and T. Vejda, "Trusted computing for the Java™platform," 2009. [Online]. Available: http://trustedjava.sourceforge.net/

[17] A. Kivity, V. Kamay, D. Laor, U. Lublin, and A. Liguori, "kvm: the Linux Virtual Machine Monitor," in *OLS2007: Proceedings of the Linux Symposium*, 2007, pp. 225–230.

[18] International Organization for Standardization (ISO), "ISO/IEC 14443: Identification Cards - Contactless Integrated Circuit(s) Cards - Proximity Cards," 2000.

[19] NFC Forum, "NFC Forum Type 4 Tag Operation - Technical Specification," NFC Forum, March 2007.

[20] National Institute of Standards and Technology (NIST), "FIPS-186-2: Digital Signature Standard (DSS)," January 2000. [Online]. Available: http://www.itl.nist.gov/fipspubs/

[21] T. Izu, B. Möller, and T. Takagi, "Improved Elliptic Curve Multiplication Methods Resistant against Side Channel Attacks," in *INDOCRYPT*, ser. Lecture Notes in Computer Science, A. Menezes and P. Sarkar, Eds., vol. 2551. Springer, 2002, pp. 296–313.