Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Vol I
IMECS 2008, 19-21 March, 2008, Hong Kong

Trends and Directions in Trusted Computing: Models, Architectures and Technologies

Muhammad Amin[1], Shabaz Khan[2], Tamleek Ali[3], Saleem Gul[4]

***Abstract*—Until recently, all the security measures have addressed servers or networks while clients or network endpoints have missed the required security concerns relatively. Most of the mechanisms safeguarding endpoints (clients) are software based. Making endpoints survive in open and reasonably exposed environments-like internet-demand that client security should stand by a tried and true dependence and merely software based mechanisms are inadequate in providing the desired security level. Trusted Computing (TC) initiatives solve these security problems through operating environments, applications and secure hardware changes to the personal computer. Using secure hardware as a basis for trusted computing provides a level of relevance since hardware-based security is mooted difficult to compromise than conventional approaches. Therefore, TC provides a powerful set of features to implement applications such as secure auctions, integrity measurement, and biometric identifcation. In this paper, we present a detailed discussion of the different approaches towards a trusted computing platform; examine these approaches and provide a set of attack mechanisms enforced if trusted initiatives are not employed at user-level. The future directions along this line of research are also conferred.**

Keywords: Network Security, Trusted Computing

I. Introduction

Network security is the key factor in the evolution of the computer software architecture and eventually trusted technology systems. Since the Network endpoint security has remained exposed comparatively and the entire security bases has been dependent on operating systems or high level security suites as convention, serious ramifications were always due to jeopardize the measures guarding these endpoints. To fill the gap, the Trusted Computing Platform Alliance (TCPA) [1] was formed in late 90's and emerged as trusted computing group (TCG) in 2003.

As a remedy to software only approaches the TCG eventually proposed secure and trustworthy setup comprehending existing capabilities, like the X.509 standard for digital certificates, IPSEC (Internet Protocol Security Protocol), IKE (Internet Key Exchange), VPN (Virtual Private Network), PKI (Public Key Infrastructure), PC/SC Specification for smart cards, biometrics, S/MIME (Secure Multi-purpose Internet Mail Extensions), SSL, SET (Secure Electronic Transaction), IEEE 802.11 WEP, IEEE 802.1x, etc [2].

TCG's primary work is the development of an inexpensive chip, known as the Trusted Platform Module (TPM) [3] or the secure hardware, that can help users protect information assets from compromise due to external software attack through integrating security standards mentioned above, at the hardware level. A software based TPM [4] for testing and research is also available.

TPM verifies the system integrity in network environment, and turns a system into a trusted one through Core Root of Trust Measurement (CRTM) including trusted boot, strong process isolation and remote attestation. The trusted boot model refers to measuring the control transfer from BIOS to the boot loader and finally booting the trusted OS. Strong isolation provides support to processes against manipulation by other processes. Virtualization [5] techniques can also address this issue. Remote attestation [6] is the key process for satisfying and verifying the authenticity of the platform.

TPM uses 2048 bit RSA for encryption/decryption, SHA-1 hashes and random number generator. Two types of keys are required: Endorsement Key (EK) is a pair of public-private key, Attestation Identity Key (AIK) is used for system authenticity which essentially differs from user authentication. Endorsement Certificate and Platform Certificates assure the integrity of EK and secure components respectively.

Another variant of secure hardware is secure coprocessor [7] and is not specified by TCG. A secure coprocessor is a hardware module containing a CPU, bootstrap ROM, and secure non-volatile memory. This hardware module is physically shielded from penetration, and the I/O interface to the module is the only way to access the internal state of the module.

Figure 1 shows the TC architecture. The architecture is composed of software and hardware parts – the latter of which is composed of the TPM and Virtualization modules.

Section I presented the inroduction. In section II, we present the evolution of trusted hardware assisted security which ultimately led to the design and implementation of TCG specific solution. Section III, discusses the modern industry standards and section IV describes the research models and archtectures. Finally, we present the threat model and strength and weaknesses of TC. In the end, future work and conclusion is also conferred.

[1]M. Amin is with Institute of Management Sciences, Peshawar, Pakistan. (e-mail: clickforamin@gmail.com, Phone: 92 0321 9033959).

[2]S. Khan is with Institute of Management Sciences, Peshawar, Pakistan. (e-mail: sahzalive@gmail.com).

[3]T. ali is with Islamic International University, Islamabad, Pakistan. And on leave from Institute of Management Sciences, Peshawar, Pakistan. (e-mail: tamleek@iiu.edu).

[4]S. Gul is with Institute of Management Sciences, Peshawar, Pakistan. (e-mail: sguls2@gmail.com).

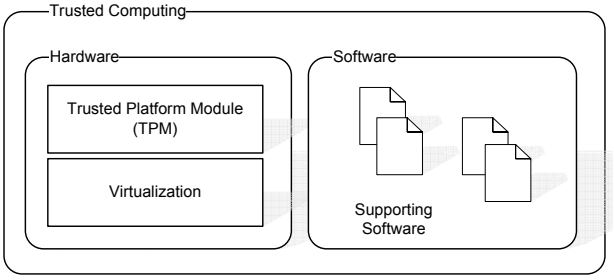

Fig. 1. Trusted Computing Architectur Overview

II. Prior Work

A. KENT

The early work emerged for software protection using secure hardware by best in 1970s. This work was taken up by Kent [8] and enhanced by providing a mechanism for encrypted storage. The original proprietary software was installed in the secure hardware through encrypted storage which was then computed [9].

B. ABYSS

In 1990 an effort by Steve white and Lian comerford, again proposed a model for software protection as A Basic York Town Security System (ABYSS) [10]. The model was secure hardware dependent. ABYSS worked in a partitioned environment where computations were carried out within the secure hardware and the rest of the normal or unprotected system communicated with that [9].

C. CITADEL

ABYSS Model was further carried on by White et al [11], [12] who proposed Citadel. The citadel improved the ABYSS protecting only secure software. Citadel provided security basis for the rest of the unprotected system. Many of the prototypes emerged from their model [9].

D. DYAD

Yee and Tygar at Carnegie Mellon University used early security architectures and implemented a system as DYAD [13], Based on secure coprocessors and support for microkernel to run. Virtual memory was made available for secure applications. All the communication to and from the coprocessor was encrypted and decrypted [9].

In the next section we present the current industry standards and approaches towards trusted computing.

III. Modern Technologies and frameworks

A. IBM

The concept of applying secure hardware was first introduced by IBM. To enable trust in system execution, IBM 4758 [14],[15] started as a research project to build a secure, tamper-resistant cryptographic coprocessor. IBM 4758 is widely used and its recently available successor – the IBM 4764 strengthens the security needs further.

Both the coprocessors 4758/4764 enables the users to prove its integrity to remote systems through attestation, which serves as the core of crypto processors [16].

Sailer et al leverages TCG in Integrity Measurement Architecture (IMA)[3], to enhance the role of the TPM not only to measure static state of a system but also dynamic state. IMA is implemented as a Linux Security Module [17] to measure each executable, library, or kernel module upon loading and record the SHA-1 hash values into TPM and the log.

B. AEGIS

A group at MIT Proposed The AEGIS [18] architecture mainly for Digital Rights management (DRM) and secure/integrated execution of programs. AEGIS initiates a mechanism which transforms a CPU in to a Trusted Computing Platform (TCP). Since the CPU contains the secure secret, all the adversaries roam around the CPU. Consequently, a secure mode of execution is added to the CPU. The memory protection is addressed at core level, i.e. a particular memory area running a program is protected or the complete memory is left unprotected.

C. XOM

David et al at Stanford proposed Execute Only Memory (XOM)[19] addressing software protection to put an end to piracy. In XOM with defined changes, the programs used to be encrypted using a key and the attorney factor in decryption remained the CPU. Because CPU had the decryption key, the programs needed to depend and trust CPU only for their integrity and protection.

D. INTEL-LaGrande/Trusted Execution Technology (TXT)

Trusted Execution Technology (TXT) [2], [20], [21] a set of enhanced hardware components designed to help protect sensitive information from software-based attacks. Trusted Execution features include capabilities in the microprocessor, chipset, I/O subsystems, and other platform components. When coupled with an enabled operating system and enabled applications, Trusted Execution Technology can help protect the confidentiality and integrity of data in the face of these increasingly hostile security environments.

The innovations of TXT include domain separation, protected execution, sealed storage, a trusted channel to graphics and input devices, mechanisms for authenticating the launch of a protected environment, attestation of platform identity and protected launch. These building blocks will help enable a new, open, software architecture for security that can help protect a user's or corporation's sensitive information from software based attacks.

E. Microsoft-NGSCB

Microsoft Next generation secure computing base (NGSCB)[22] operates in two modes i.e. Normal and Trusted modes. Normal mode is unprotected and controlled by the users, while trusted mode is the implementation of TC.

TABLE I
TPM FEATURE SET

Type of control	User Managed
Location/Binding	Hard wired
No. of OSs supported	1
Architechture	Centralized
Popular OSs Supported	Windows Vista/Linux
Library/Software stack	TPMd/TrouSers
Leading Initiative	NGSCB, TXT, PRESIDIO
Prominent TPM Assissted Frameworks	Secure Electronic Transction(SET), SAM, SHEMP
Enhancements Through TPM	S-MIME e-mail, VPN and PKI Authentication and Wireless Authentication for 802.1x
Providers	Atmel, Broadcom, Infineon

NGSCB has two main components namely *NEXUS* and *Nexus Computing Agent (NCA)*.

Nexus is a special kernel for providing process isolation between normal and trusted modes. It also carries out authentication and encryption of data. NCA is trusted mode software which is responsible for communicating with Nexus kernel.

The main features offered by NGSCB are strong process isolation, sealed storage, attestation and secured path to users with the help of TPM.

F. HP ProtectTools

HP ProtectTools [23] Embedded Security is a hardware security chip – TPM – that integrates the core elements of trust into the system. This embedded security initiative provides simple file and folder encryption and flexible platform management. The main feature incorporated is unique key encryption and storing it on highly uncompromised silicon storage.

The support for wireless users authentication and protection, strengthened email system, and control over systems connecting to network for limiting rights access are robust feature of Hp ProtectTools.

HP-UX Trusted Computing Services (HP-UX TCS) [24] provides software support for hardware-enforced key management on supported HP Integrity servers running HP-UX 11iv2. HP-UX TCS is primarily composed of the following elements:

TPM drivers for kernel, TCG specific software stack, maintenance utilities, commands for user specific encryption/decryption and secure storage.

G. AMD Presidio

AMD's LaGrande equivalent is Presidio [25]. AMD's Presidio is initiative towards trusted computing which extends its predecessor effort of Enhanced Virus Protection (EVP). The main objective is achieved through certain hardware changes to processor and chipset features. The secure partitioning and secure input/output paths have been seriously looked up by employing a concept such as TPM. Isolated execution space, Enhanced Virus Protection (EVP), storage sealing and remote attestation etc are few capability examples offered by AMD Presidio.

H. DELL

Dell includes TPMs and Wave Systems EMBASSY Trust Suite software on many Dell Latitude notebooks, Dell OptiPlex desktops, and Dell Precision workstations [26]. Dell also anticipates eventually incorporating TPM architectures on its servers and storage.

I. Apple

Apple does not appear to have plans to use Trusted Computing [27], [28]. It was planned that Mac OS X will use the TPM when available (on the Intel Mac platform) to protect the OS from piracy. Although Open-Source Software (OSS) support is available, based on the FreeBSD UNIX system, but it is not yet distributed by Apple [27].

J. ARM-Trust Zone

TrustZone [29]is a security extension introduced by ARM Ltd. in its ARM 1176 core. ARM is well known for 32-bit Embedded CPUs, mainly used for PDAs and mobile phones. TrustZone is a technology to provide a hardware-based security for the security critical part of the sytem rather than maintaining the whole sytem in secure state all the time [30]. The security-sensitive applications are run executed in a separate memory space which is not accessible to normal applications. TrustZone is the first ARM architecture to provide hardware based security in its core [31].

Table II shows a comparison of different hardware architectures from the perspective of their goals and different approaches.

IV. RESEARCH MODELS AND ARCHITECTURES

A. SAM: A Flexible and Secure Auction Architecture Using Trusted Hardware

Secure Auction Marketplace (SAM) [32] mitigates challenges faced by auctioneer/bidders to conduct auctions efficiently. SAM enables the bidders to experience correct bidding result, confidentiality of bids, and anonymity without being exposed and trusting the auctioneer. Since auctioneer could use the auctioning system in their favor.

SAM addresses this problem using secure coprocessor environment such as IBM 4758. SAM components such as auction controller and bid collector are loaded onto the secure coprocessor securely. The trusted software and the secure coprocessor together act as a secure auction marketplace (SAM). SAM becomes an authenticated computational entity, whose internal state and operations cannot be examined or altered as all the auction specifications, bids and results are held and evaluated by SAM; an adversary even one with direct physical access to that hardware can't compromise the security measures.

B. A Trusted Biometric System

Chen et al [34] described a method for biometric identification based user authentication in distributed environments. The main entities in the system are a user, a trusted platform

TABLE II
COMPARISONS OF HARDWARE ARCHITECTURES (EXTENDED FROM [33])

	XOM, AEGIS, CITADEL	**TCG, NGSCB, TXT, Presidio, ProtectTools**	**ARM**
Goal	Copy and tamper-resistant software distribution and execution	Copy-resistant software distribution	Embedded systems
Require separate hardware?	No	Yes	Yes
Require permanent device secret?	Yes	Yes	Yes
Source of user's trust	Integrity of S/W and computation results	Integrity of S/W at load time	H/W based separation of secure S/W
Trust ties users to devices	Yes	Yes	Yes
Security Perimeter	Processor chip boundary	Procesor, DRAM, chipset etc.	Microprocessor
Targets embedded systems	No	No	Yes

using a TPM, smartcard (SC) and a Trusted Biometric Reader (TBR). The system is supposed to provide a trust establishment mechanism between the user and the biometric reader. The user is presented with an integrity report of biometric mechanism through the trusted system, only then user is to trust the reader and release his/her sensitive information.

C. SHEMP: Secure Hardware Enhanced MyProxy

MyProxy is an online credential repository developed by Grid community for providing security and mobility of sensitive records of users [35]. Secure Hardware Enhanced MyProxy (SHEMP) [36] was presented as an extensive architecture for MyProxy by providing strong security basis supported with secure coprocessors using different hardware at client and repository. Proxy certificates (PCs) were integrated with applications for un-attackable authentication.

Client and repository properties were managed through eXtensible Access Control Markup Language (XACML) to provide users with flexibility for specifying key usage options. SHEMP have been prototyped and tested. The repository and a client run on trusted Bear [37].

D. Satem: A Trusted Computing System

A Service-aware Attestation Method (Satem) [38] Towards Trusted Service Transaction is composed of a trusted agent in the OS kernel of the service platform and a trust evaluator on the user platform assures the management and trusted execution of the code in Ad-hoc networks. Trusted boot process is carried out to attest the OS kernel and trusted agent by the service platform using TPM.

Satem uses trusted boot to establish trusted computing base incorporating trusted agent as well as the entire OS kernel. Where no component in the boot sequence can bypass the any other without verifying its integrity and authenticity, thus each component attests the next one before handing over the control. A same like system is also used in satem; Being a linux based system all the boot attestation results are stored in TPM. Since TPM computes the SHA-1 hash over the BIOS image, BIOS computes the SHA1 hash over Linux Loader (LILO), and LILO does the same for the OS kernel. This helps the OS kernel and agent to be trusted and proves the system to be genuine.

E. EMSCB: European Multilaterally Secure Computing Base

EMSCB [27] is a German project involving Infineon, SAP, Blaupunkt, Sirrix, Universities of Bochum and Dresden. It is based on the PERSEUS[39] Security Framework and the L4 [40] microkernel.

EMSCB is actively pursuing the development of a secure kernel for common usage [27]. And Turaya is the security architecture by EMSCB. PERSEUS and L4 serves as its basis. Integrity of security Kernel is verified by the TPM [41]. Two prototype applications have been implemented:

- Disk encryption (Turaya.Crypt)
- VPN communication (Turaya.VPN)

Turaya.Crypt implements a device encryption module and provides transparent encryption scheme from users. It is mainly used for removable devices like USB [41].

Turaya.VPN implements a tranparent and secure VPN communication and encryption module. which is based on IPSec [41].

F. Minimal TCB Code Execution

McCune et al [42] have proposed a Secure Execution Architecture (SEA) with a dramatic decrease in trusted computing base (TCB) code which efficiently isolates the sensitive code from unnecessary process intervention. SEA takes advantage of AMD's SVM and Intel's TXT since these platforms guarantee hardware based root of trust and security measures without a system reboot.

In the next section, we present the basic threat models in cases where TC is not implemented.

V. ADVERSARY MODEL

A. Physical Attack

Refers to direct physical access to security devices or there inner components. For example, the attacker may open a hole in the passivation layer of a microcontroller chip and place a micro-probing needle on a bus line in order to capture a signal [43].

Hardware assisted security used in 1980s was highly exposed to physical attacks. As an attacker expert in precision drilling could make his way through. Since batteries were used to hold crypto information and were required to be changed for maintenance, technical personnel operating these batteries for routine checkup could easily uncover the secure information held in. IBM 4758 came up with a solution by placing these batteries outside the secure primitives and deploying temper resistant membranes on top of it.

Secondly, the cost of technology helps avoid physical attacks while these devices are becoming more complex and

TABLE III
TPM STRENGTHS & CHALLENGES

Strengths & Challenges	**TPM Concern**	**TPM Prposed solution**
S/W Attack	√	–
Platform Integrity	√	CRTM/Remote Attestation
H/W Attack	×	–
AES Encryption	×	–
Confidentiality of data& S/w	√	Credential Encryption/hash calculation
Integrity of data& S/w	√	Credential Encryption/hash calculation
Certificate Mgt.	√	Verified by CA
Unauthorised N/W access	√	H/W prorection of sensitive data
VPN Support	√	Trusted Network Connect(TNC)
Key Security	√	RSA Encryption
Platform Verification	√	SHA–1 Hashes
Unique Nonce/Key Generation	√	RNG
TPM Integrity	√	Endorsement Keys
Network Identity	√	Trusted Network Connect(TNC)
Digital Rights management(DRM)	×	TPM does not directly address DRM
Anti-Worm Protection	×	–

expensive. This excludes a major class of active attackers. Although TPM is active in providing crypto processing, it is yet not been able to provide protection against the physical attacks.

B. Remote Attack

Types of remote attacks include timing analysis, cryptanalysis, protocol analysis and attacks on application programming interfaces. Cryptanalysis and protocol analysis are well known attacks where the security threats exploit the flaws in design such as hash algorithms in the former case and in the latter case, uses the flaws in the systems which uses these crypto standards and designs [43].

The TPM can protect against attacks on software integrity only. If an adversary compromises a system, BIOS or the critical information residing on the physical memory, TPM will not disclose the secrets.

There is another class of attacks permitted by TPM barring additional counter measures. Suppose a protected object has value v_0 at time t_0 and $v_1 = v_0$ at time $t_1 > t_0$. If the adversary makes a copy of the hard disk at time t_0, the adversary can restore the value v_0 by powering down the system and loading the old copy. For some applications, this attack can have serious ramifications (e.g., it might permit the adversary to restore revoked privileges or spent e-cash, or roll back a security-critical software upgrade)[44].

VI. STRENGTHS AND CHALLENGES OF TRUSTED COMPUTING

Ever increasing data security need requires challenging security hardware. Systems protected with TPMs will not only protect data but also keep the integrity of the data uncompromised. TPM limits the access to the data–from any threat or an individual from an open environment like internet seeking access to the data stored–without the provision of original credentials . TPM also encrypts the data stored, even an un-authorised software attempting to access the secure data will be blocked- sealed off the memory [45].

Since the TPM cannot differentiate a potential user or an adversary, this poses a serious challenge. Any potential user seeking access to the secure system will also be treated as a threat. User failing to provide or losing credentials or forgetting a password can have dire consequences. [45]. Table III summarizes these strengths and challenges.

VII. FUTURE WORK

TPM enabled operating systems and other softwares are currently under development and will be available next year, which will take the data security to an entirely new level. It can be clearly viewed that high processing capabilities at PC level like Quad and Octa cores from Intel and AMD may boost cryptographic computations that ultimately can effect the TPM's embedded in the systems. Also the TPM performance is related to upcoming Virtualization techniques, thus, if the size and complexity of the virtual machines and trusted computing code can be substantially decreased it will have better performance results as in [42]. The most recent work is to from [31] as they have proposed MAC and selinux for trusted embedded systems using ARM core.

VIII. CONCLUSION

This paper presented a concise overview of trusted computing initiative and its evolution. A brief summary of modern industry standards and models are described. Finally, TCG specific applications and general threat model is explained. Thus, TPM based security measures will become more efficient and widespread with increase in computation power and reduction in size and complexity of virtualization technologies.

REFERENCES

[1] "Trusted Computing Group," *Available at: http://www.trustedcomputinggroup.org*, 2003.
[2] S. Bajikar, "Trusted Platform Module (TPM) based Security on Notebook PCs-White Paper," *White Paper, Mobile Platforms Group–Intel Corporation*, vol. 20, 2002.
[3] "Trusted Platform Module Main Specification," *Trusted Computing Group. Available at: http://www.trustedcomputinggroup.org*, 2003.
[4] M. Strasser, "A Software-based TPM Emulator for Linux," *Semester Thesis, ETH Zurich*, 2004.

[5] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," *Proceedings of the nineteenth ACM symposium on Operating systems principles*, pp. 164–177, 2003.

[6] T. Garfinkel, M. Rosenblum, and D. Boneh, "Flexible OS Support and Applications for Trusted Computing," *Proceedings of the 9th Workshop on Hot Topics in Operating Systems, Kauai, Hawaii*, 2003.

[7] B. Yee and J. Tygar, "Secure coprocessors in electronic commerce applications," *Proceedings of The First USENIX Workshop on Electronic Commerce, New York, New York, July*, 1995.

[8] S. Kent, "PROTECTING EXTERNALLY SUPPLIED SOFTWARE IN SMALL COMPUTERS," 1981.

[9] S. Smith, "Magic boxes and boots: security in hardware," *Computer*, vol. 37, no. 10, pp. 106–109, 2004.

[10] S. White and L. Comerford, "ABYSS: an architecture for software protection," *IEEE Transactions on Software Engineering*, vol. 16, no. 6, pp. 619–629, 1990.

[11] S. White, S. Weingart, W. Arnold, and E. Palmer, "Introduction to the Citadel Architecture: Security in Physically Exposed Environments," *Technical ReportRC16672, Distributedsecurity systems group, IBM Thomas J. Watson Research Center, March*, 1991.

[12] E. Palmer, *An Introduction to Citadel: A Secure Crypto Coprocessor for Workstations*. IBM TJ Watson Research Center, 1992.

[13] J. Tygar and B. Yee, *Dyad: A System for Using Physically Secure Coprocessors*. School of Computer Science, Carnegie Mellon University, 1991.

[14] J. Dyer, M. Lindemann, R. Perez, R. Sailer, L. van Doorn, and S. Smith, "Building the IBM 4758 secure coprocessor," *Computer*, vol. 34, no. 10, pp. 57–66, 2001.

[15] S. Smith and S. Weingart, "Building a high-performance, programmable secure coprocessor," *COMPUT. NETWORKS*, vol. 31, no. 8, pp. 831–860, 1999.

[16] "Research for Advancing Trusted Computing," *Available at: http://domino.research.ibm.com/research.nsf/pages/r.security.innovation.html*, 2007.

[17] C. Wright, C. Cowan, J. Morris, S. Smalley, and G. Kroah-Hartman, "Linux security modules: general security support for the linux kernel," *Foundations of Intrusion Tolerant Systems, 2003 [Organically Assured and Survivable Information Systems]*, pp. 213–226, 2003.

[18] G. Suh, D. Clarke, B. Gassend, M. van Dijk, and S. Devadas, "AEGIS: architecture for tamper-evident and tamper-resistant processing," *Proceedings of the 17th annual international conference on Supercomputing*, pp. 160–171, 2003.

[19] D. Lie, C. Thekkath, M. Mitchell, P. Lincoln, D. Boneh, J. Mitchell, and M. Horowitz, "Architectural support for copy and tamper resistant software," *ACM SIGPLAN Notices*, vol. 35, no. 11, pp. 168–177, 2000.

[20] "Intel Trusted Execution Technology (TXT)," *Intel Corporation. D52212*.

[21] L. van Doorn, "Hardware virtualization trends," *Proceedings of the 2nd international conference on Virtual execution environments*, pp. 45–45, 2006.

[22] M. Peinado, Y. Chen, P. England, and J. Manferdelli, "NGSCB: A Trusted Open System," *Proceedings of 9th Astralasian Conference on Information Security and Privacy, ACISP*, vol. 3108, pp. 86–97, 2004.

[23] "Embedded Security for HP ProtectTools," *Available at: http://h20331.www2.hp.com/Hpsub/cache/292199-0-0-225-121.html*, 2007.

[24] "HP-UX Trusted Computing Services Release Notes HP-UX 11i v2," *Available at: docs.hp.com/en/5992-0553/5992-0553.pdf, March*, 2007.

[25] G. Strongin, "AMDs Vision for Trustworthy Computing," *Available at: conference.digitalidworld.com/2004/attendees/slides/1027_1700_E1.pdf, October*, 2004.

[26] F. MOLSBERRY and B. BERGER, "Enhancing IT Security with Trusted Computing Group Standards," *Available at: www.dell.com/downloads/global/vectors/2007_tcgs.pdf*, 2007.

[27] "Trusted Computing Applications," *Available at: http://www. isg. rhul. ac. uk/files/ IY5608__Lecture_7_Operating_Systems. pdf*, 2007.

[28] A. Singh, "Trusted Computing for Mac OS X," *Available at: http://www.osxbook.com/book/bonus/chapter10/tpm/*, 2006.

[29] T. Alves and D. Felton, "Trustzone: Integrated hardware and software security," *ARM white paper, July*, 2004.

[30] K. Dula, R. Manolakos, and Zalewski, "Security of Hardware Security-Homework Presentation TrustZone by ARM," *Available at: soc.eurecom.fr/crypto/Presentations/2006-2007/HWsec_TrustZone.pdf, January*, 2007.

[31] H. Nahari, "Trusted Secure Embedded Linux: From Hardware Root Of Trust To Mandatory Access Control," *Proceedings of the Linux Symposium*, pp. 79–85, 2007.

[32] A. Perrig, S. Smith, D. Song, J. Tygar, and U. Berkeley, "SAM: a flexible and secure auction architecture using trusted hardware," *Parallel and Distributed Processing Symposium., Proceedings 15th International*, pp. 1764–1773, 2001.

[33] R. Lee, P. Kwan, J. McGregor, and J. Dwoskin, "Architecture for protecting critical secrets in microprocessors," *Computer Architecture, 2005. ISCA'05. Proceedings. 32nd International Symposium on*, pp. 2–13, 2005.

[34] L. Chen, S. Pearson, and A. Vamvakas, "A Trusted Biometric System," Technical Report HPL-2002-185, Hewlett-Packard, 2002, http://www. hpl. hp. com/techreports/2002/HPL-2002-185. html, Tech. Rep., 2002.

[35] J. Novotny, S. Tuecke, and V. Welch, "An Online Credential Repository for the Grid: MyProxy," *Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10)*, pp. 104–111, 2001.

[36] J. Marchesini, "SHEMP: Secure Hardware Enhanced MyProxy," Ph.D. dissertation, DARTMOUTH COLLEGE, 2005.

[37] J. Marchesini, S. Smith, O. Wild, and R. MacDonald, "Experimenting with TCPA/TCG Hardware, Or: How I Learned to Stop Worrying and Love The Bear," *Computer Science Technical Report TR2003-476, Dartmouth College*, 2003.

[38] G. Xu, C. Borcea, and L. Iftode, "Satem: Trusted Service Code Execution across Transactions," *Proc. IEEE Int. Symp. Reliable Distributed Systems*, 2006.

[39] B. Pfitzmann, J. Riordan, C. Stuble, M. Waidner, and A. Weber, "The PERSEUS system architecture," *VIS*, pp. 1–18, 2001.

[40] K. Elphinstone, "Future directions in the evolution of the L4 microkernel," *Proc. NICTA Formal Methods Workshop on Operating Systems Verification. National ICT Australia*, 2004.

[41] "European Multilaterally Secure Computing Base," *Available at: http://www.emscb.com/content/pages/turaya.htm*, 2007.

[42] J. McCune, B. Parno, A. Perrig, M. Reiter, and A. Seshadri, "Minimal TCB Code Execution," *Security and Privacy, 2007. SP'07. IEEE Symposium on*, pp. 267–272, 2007.

[43] R. Anderson, M. Bond, J. Clulow, and S. Skorobogatov, "Cryptographic Processors-a survey," *Proceedings of the IEEE*, vol. 94, no. 2, pp. 357–369, 2006.

[44] R. MacDonald, S. Smith, J. Marchesini, and O. Wild, "Bear: An Open-Source Virtual Secure Coprocessor based on TCPA," *Computer Science Technical Report TR2003-471, Dartmouth College, August*, 2003.

[45] S. Chun, "Implementation of trusted compuring," *Available at: www.techteam.com/governmentsolutions/images/it%20october%2006%203.pdf*, 2006.