Purchase Order US001-0000328568

Battelle Memorial Institute
505 King Ave.
COLUMBUS OH 43201-2696

Change Order	4		Dispatch via E-Mail
Date	**Contract ID:**	**Release:**	**Page**
05/09/2012			1 of 5
Payment Terms	**Freight Terms**		**Ship Via**
NET 30	FOB DEST, Freight PP&Add		Common
Applicable Purchase Order Supplements			

UNIVERSITY OF KANSAS
CENTER FOR RESEARCH INC
ACCOUNTS PAYABLE
2385 IRVING HILL ROAD WEST CAMPUS
LAWRENCE KS 66045-7563

Service Performed At: 651 West Fifth Ave.
COLUMBUS OH 43201-3174

Bill To: Attn: Accounts Payable
505 King Ave.
COLUMBUS OH 43201-2696

Procurement Notes

THIS SUBCONTRACT is issued to the Subcontractor identified above by BATTELLE MEMORIAL INSTITUTE ("Battelle"), a charitable trust organized as a non-profit corporation under the laws of the State of Ohio, in support of its internal R&D activities..

WHEREAS, Battelle desires Subcontractor to perform services as more fully described in Article II below;

WHEREAS, Subcontractor is willing to undertake the performance of such services on a Firm Fixed Price basis; and Battelle finds Subcontractor is qualified to perform such services, all relevant factors considered.

NOW, THEREFORE, in consideration of these premises, the parties do mutually agree to the following:

I. PERIOD OF PERFORMANCE - The period of performance for this Subcontract is identified on each respective line item of this Subcontract. The Subcontractor is not authorized to perform any work under this Subcontract beyond the performance period set forth above unless such period is extended by written modification to this Subcontract.

II. SCOPE - Subcontractor shall, on the terms and conditions attached hereto, furnish the necessary management, qualified personnel, facilities, equipment and materials necessary for and incidental to the performance of the services set forth in "A Proposal for TPM Verification Dr. Perry Alexander, The University of Kansas March 27, 2012.

III. INVOICING/PAYMENT - In consideration of Subcontractor's responsibilities under this Subcontract, Battelle shall pay Subcontractor two firm fixed price payments identified below and upon presentation of invoices in accordance with Standard University of Kansas approved process.

IV. TOTAL Firm Fixed Price - The total Firm Fixed Price of the Subcontract is $76,551 and cannot be exceeded except with the express approval of Battelle Memorial Institute.

V. INVOICES: The University of Kansas shall provide two invoices to satisfy to contractual deliverables as follows:

May 30, 2012 Written Summary Report Deliverable Payment Amount $38,275.50
September 7, 2012 Written Summary Report Deliverable Payment Amount $38,275.50

The University Shall submit its September 7, 2012 invoice no later than close of business on September 7, 2012 and recognizes that delivery of this invoice is critical and therefore time is of the essence for this submission.

VI. ACCOUNTING RECORDS - All direct costs incurred on this Subcontract shall be charged by Subcontractor to accounts that are separate from all others within Subcontractor's accounting records/system and are for the express purpose of collecting costs incurred for this effort.

VII. NOTIFICATION - Whenever Subcontractor has reason to believe that the total price of the work under this Subcontract will be greater or substantially less than the amount authorized, Subcontractor shall promptly notify the Battelle Subcontracting Officer. No additional funds are authorized beyond the stated price ceiling, unless modified by both parties in writing.

VIII. WITHHOLDING - Battelle may withhold payment of any invoice if Subcontractor has not complied with any material requirement of this Subcontract. Said payment will be paid only when the requirement is satisfactorily met. Any payment so withheld shall not accrue interest.

IX. TRAVEL – Payment for authorized travel costs shall be in accordance with the applicable requirements of the University of Kansas Travel Policy.

Purchase Order US001-0000328568

Battelle Memorial Institute
505 King Ave.
COLUMBUS OH 43201-2696

Change Order	4		Dispatch via E-Mail
Date 05/09/2012	**Contract ID:**	**Release:**	**Page** 2 of 5
Payment Terms NET 30	**Freight Terms** FOB DEST, Freight PP&Add		**Ship Via** Common
Applicable Purchase Order Supplements			

UNIVERSITY OF KANSAS
CENTER FOR RESEARCH INC
ACCOUNTS PAYABLE
2385 IRVING HILL ROAD WEST CAMPUS
LAWRENCE KS 66045-7563

Service Performed At: 651 West Fifth Ave.
COLUMBUS OH 43201-3174

Bill To: Attn: Accounts Payable
505 King Ave.
COLUMBUS OH 43201-2696

Procurement Notes

1. Submit invoices via email (preferred) to: accountspayable@battelle.org and also to ritondom@battelle.org Battelle Memorial Institute, Columbus Operations, 505 King Avenue, Columbus, Ohio 43201-2693, Attention: Accounts Payable. Invoices shall reflect the Subcontract number. Invoice terms are net 30 days. The final invoice should be marked "Final Invoice".

2. This Subcontract may be terminated, in whole or in part, by 30 days written notice of either party for any reason. The notice of termination shall specify the extent to which performance is terminated and the effective date of such termination. Subcontractor shall be reimbursed for all actual and allowable expenses and all uncancellable obligations properly incurred prior to the date of termination.

3. Battelle may at any time, by written order, require the Subcontractor to stop all, or any part, of the work. Upon receipt of such an order the Subcontractor shall take all reasonable steps to minimize the incurrence of costs allocable to the work covered by the order during the period of work stoppage. Battelle shall either: (i) cancel the stop work order, or (ii) terminate the work covered by such order.

4. Subcontractor and its employees shall maintain in strict confidence all information received from Battelle including, but not limited to Battelle client information, specifications, business and market plans & procedures, test plans, protocols, test results, results of analyses, project notebooks, project documentation, notebooks, and other technical, business, and trade secret information.

5. Subcontractor shall assume all risks of personal injury, including death, property damage or other loss caused by its or its employee or agents own negligent acts or omissions.

6. Intellectual Property
Definitions
"Foreground Intellectual Property" shall mean inventions, discoveries and works of authorship subject to protection by patent, copyright or any other intellectual property rights in the United States or elsewhere, and technical information and know-how, that are conceived or otherwise first created in the performance of the Project by one of the parties to this agreement or jointly by the parties during performance of the Project.
"Background Intellectual Property" shall mean inventions, discoveries and works of authorship subject to protection by patent, copyright, or any other intellectual property rights, in the United States or elsewhere, and technical information and know-how, owned by one of the parties to this Agreement (or which one of the parties to this Agreement has the right to use for the Project) that are conceived or documented or otherwise first created and documented outside the performance of this Project.

(continued on the next page)

Purchase Order US001-0000328568

Battelle Memorial Institute
505 King Ave.
COLUMBUS OH 43201-2696

Change Order	4		Dispatch via E-Mail	
Date	**Contract ID:**		**Release:**	**Page**
05/09/2012				3 of 5
Payment Terms	**Freight Terms**			**Ship Via**
NET 30	FOB DEST, Freight PP&Add			Common
Applicable Purchase Order Supplements				

UNIVERSITY OF KANSAS
CENTER FOR RESEARCH INC
ACCOUNTS PAYABLE
2385 IRVING HILL ROAD WEST CAMPUS
LAWRENCE KS 66045-7563

Service Performed At: 651 West Fifth Ave.
COLUMBUS OH 43201-3174

Bill To: Attn: Accounts Payable
505 King Ave.
COLUMBUS OH 43201-2696

Procurement Notes

Ownership and Allocation of Rights of Foreground Intellectual Property
Title: Title to Foreground Intellectual Property solely developed by University will vest solely in University. Title to Foreground Intellectual Property developed solely by Battelle will vest solely in Battelle. Title to Foreground Intellectual Property jointly developed by University and Battelle will vest jointly in University and Battelle. Title to each party's Background Intellectual Property will remain the exclusive property of the party who owns the Background Intellectual Property.
Disclosure: University will promptly notify Battelle in writing of any and all Foreground Intellectual Property conceived or otherwise first created in the performance of the Project.
Internal Use License: University grants Battelle a paid-up, royalty free, nonexclusive right to practice all Foreground Intellectual Property solely developed by University for Battelle internal purposes (including specifically, research and development work for Battelle and for its R&D funding clients).
Option for Commercial license (Foreground Intellectual Property): Battelle shall advise University within a period of six (6) months from the date of completion of the Project or any extensions or modifications thereof whether or not it wishes to secure a commercial license. If Battelle elects to secure a commercial license, Battelle shall assume all costs associated with securing and maintaining patent protection for such invention(s), whether or not Letters Patent issue. Upon election, the parties will negotiate a commercial license on within a commercially reasonable period of time. The license shall contain reasonable and customary terms and conditions and be negotiated and agreed to by the parties in good faith. At Battelle's election, the license may be non-exclusive or exclusive. The license shall require diligent performance by Battelle for the timely commercial development and early marketing of such inventions and include Battelle's continuing obligation to pay patent costs. If such license agreement is not concluded in said period, University has no further obligations to Battelle. If Battelle does not elect to secure such license, rights to the inventions disclosed hereunder shall be disposed of in accordance with University policies with no further obligation to Battelle.
Option for Commercial license (Background Intellectual Property): Subject to availability, the University hereby grants Battelle an option to a non-exclusive, royalty-bearing license, including the right to sublicense, to practice University's Background Intellectual Property as is necessary for Battelle to commercialize the Foreground Intellectual Property. The license shall be to make, have made, sell, offer for sale, and import or export, on reasonable and customary terms and conditions to be negotiated and agreed to by the parties in good faith. This option extends for a period of six (6) months from the date of completion of the Project or any extensions or modifications thereof.
Except as otherwise provided herein, no rights in any Background Intellectual Property shall be transferred by this Agreement; provided, however, that each party hereby grants to the other a non-exclusive, royalty-free license to use Background Intellectual Property owned by the other (or which the other party has the right to use for the Project) exclusively for the sole purpose of performing the Statement of Work for the Project.

Procurement Notes

7. Subcontractor is an independent contractor and not an employee, agent, or representative of Battelle. Subcontractor shall be solely responsible for all employment-related wages, benefits, FICA, federal and state unemployment and other taxes and payments as required by law, for itself and any persons it employs. Subcontractor shall perform the services and provide the necessary facilities, personnel, materials, equipment, and shall otherwise do all things necessary for the performance of the Statement of Work, and shall be solely responsible for its own financial obligations to third parties and to its employees and contractors. Further, Subcontractor agrees that it shall not be covered by any Battelle insurance or benefits, including but not limited to Worker's Compensation, Professional Liability, General Liability, Employer's Liability, Automotive Liability, and Unemployment Compensation. Subcontractor shall protect, defend and hold Battelle harmless from any claims or penalties asserted or assessed against Battelle by any person or governmental entity relating to Subcontractor's responsibilities under this clause.

8. Subcontractor agrees that it shall comply with all U.S. laws and regulations applicable to exports. Subcontractor agrees not to export or re-export any products, materials, items and/or technical data, or the product(s) thereof, received from Battelle unless Subcontractor has obtained, in advance, Battelle's approval and all required licenses, agreements or other authorizations from the U.S. Government. Exports include, the sending or taking of any products, materials, items or technical data that are subject to export regulations (International Traffic in Arms Regulations and/or Export Administration Regulations) out of the United States in any manner; disclosing or transferring technical data to a Foreign Person (i.e. any person who is not a lawful permanent resident of the U.S. or is not a protected individual as defined by 8 U.S.C sections 1101 and 1324) whether in the United States or abroad; or performing services for a foreign client, whether in the United States or abroad.

Subcontractor understands and agrees to comply with the United States Foreign Corrupt Practices Act, which prohibits Battelle and Subcontractor from providing anything of value to a foreign public official in order to obtain or retain business. Subcontractor agrees not to give anything of value, including but not limited to business gratuities and reimbursement of travel, to any foreign government officials. Subcontractor agrees to insure that it complies with all requirements relevant to its business arrangement with Battelle, including any registration requirements, and warrants that this Subcontract is in compliance with all applicable laws and regulations of the country or countries in which it performs any services for Battelle.

9. In the absence of the prior written approval of the other party, no public releases including those for news, or advertising, shall be issued by either party. Neither Battelle nor Subcontractor endorse products or services. Accordingly, neither party shall use or imply the other party's name or use the other party's information or reports for advertising, promotional purposes, raising of capital, recommending investments, sale of securities or in any way that implies endorsement by the other party. Acknowledgement of funding or sponsorship in a factual statement is not prohibited by this clause and Subcontractor may list the existence of this project in its internal documents, databases and annual report which is available to the public.

10. This Subcontract contains all of our understandings and agreements relating to the services and may be changed only in writing by both parties authorized representative. This Subcontract shall be governed by the laws of, and enforced within the jurisdiction of, the State of Kansas, without regard to its principles of conflicts of law.

BUSINESS ETHICS PROCEDURES: BATTELLE EMPLOYEES, SUPPLIERS, AND SUBCONTRACTORS ARE ENCOURAGED TO REPORT IN CONFIDENCE ANY APPARENT WRONGDOINGS TO THE DIRECTOR OF STRATEGIC ACQUISITION SUPPORT OR TO THE GENERAL COUNSEL (800) 201-2011.

Purchase Order US001-0000328568

Battelle Memorial Institute
505 King Ave.
COLUMBUS OH 43201-2696

Change Order	4		Dispatch via E-Mail
Date 05/09/2012	**Contract ID:**	**Release:**	**Page** 5 of 5
Payment Terms NET 30	**Freight Terms** FOB DEST, Freight PP&Add		**Ship Via** Common
Applicable Purchase Order Supplements			

UNIVERSITY OF KANSAS
CENTER FOR RESEARCH INC
ACCOUNTS PAYABLE
2385 IRVING HILL ROAD WEST CAMPUS
LAWRENCE KS 66045-7563

Service Performed At: 651 West Fifth Ave.
COLUMBUS OH 43201-3174

Bill To: Attn: Accounts Payable
505 King Ave.
COLUMBUS OH 43201-2696

Line-Sch	Item/Description	Quantity	UOM	Price/Rate	Total Amt	Start Date	Due/End Dt
1- 1	SALARIES, RESEARCH MATERIALS/COMMUNICATION COSTS/COMPUTER NETWORKING AND MAINTENANCE COSTS, INDIRECT, ON-CAMPUS RESEARCH	1.00	EA	70,833.0000	70,833.00	04/16/2012	09/30/2012
2- 1	TRAVEL TO BATTELLE COLUMBUS	1.00	EA	5,718.0000	5,718.00	04/16/2012	09/30/2012

All PO Lines are Tax Exempt

Prior Total PO Amount	76,551.00
Change Order Total	0.00
Total Estimated Sales/Use Tax	0.00
Revised Total PO Amount	76,551.00

Technical Contact: Krishnaswamy,Padma

Bilateral Signature

Joanne Altieri, Director
Research Administration

Printed Name: Joanne Altieri

Title:

Date: May 9, 2012

Battelle Authorized Signature

on behalf of
RODRIGUES,ADAM J (PROCUREMENT OFFICER)
614/424-4864 FAX 614/458-4864 rodriguesa@battelle.org

Date: May 9, 2012

A Proposal for TPM Verification

Dr. Perry Alexander, The University of Kansas

March 27, 2012

This document presents a proposal for verifying the Trusted Processor Module (TPM). We will approach the problem by developing and verifying a formal requirements model, developing a formal implementation model, and demonstrating observational equivalence between models using the PVS verification system. The anticipated outcomes are the models developed and the collection of verification results. The the short-term benefits of the effort are the models themselves while long-term benefits include substantially simpler and less expensive certification processes.

Introduction

Our proposed task is to *perform instruction level verification of a Trusted Processor Module.* To achieve this goal, the following specific tasks will be undertaken:

- *Model the TPM state and command execution* — Using monadic techniques develop a model of the TPM internal state and a framework for command execution.
- *Model and Verify the attestation subset* — Using the TPM state model, specify and verify the TPM command subset for PCR management and quote generation. Extend existing verification of PCR extension results.
- *Model and Verify the key management subset* — Using the TPM state model, specify and verify the TPM command subset for generating and managing keys.
- *Verify common TPM command sequences* — Use the new monadic model to expand previously performed verification of various TPM processes including boot

Unless directed otherwise, verification will be performed using PVS [Owre et al., 1992], SAL [Bensalem et al., 2000], and yices [Dutertre and de Moura, 2006]. It is anticipated that the vast majority of the work will be done using PVS, but we will explore defining data structures for TPM state that can be used in both PVS and SAL.

Approach

The proposed approach is a classic modeling technique common in both hardware and software verification where a concrete model is verified against an abstract model using *weak bisimulation* [Sangiorgi, 2012] or *observational equivalence* as a verification goal. An abstract model will be written and verified that defines requirements for TPM instructions. Much like a data sheet, this specification will define each command in terms of inputs, outputs, invariants, and state changes using a formal logic. Verification will ensure that individual commands satisfy invariants and requirements while command sequences accomplish their intended task without unintended consequences.

A concrete model will be written that defines implementation of the same TPM instructions. Each command will be defined formally as a transition over a concrete state representation taken from the TCG [—, 2007] definition. [1] The concrete model represents implemented instructions in contrast to the requirements represented by the abstract model.

[1] A specific implementation could be substituted for the TCG specification if desired.

Each TPM command and selected command sequences in the concrete model will be verified against its associated abstract model by proving the models are behaviorally equivalent. Specifically, the concrete model will be run on a concrete state resulting in a new concrete state. The abstract model will be run on the same state represented using the abstract state model. If the concrete state lifted into the abstract model is equivalent to the abstract model, then the two models are considered to be *weakly bisimilar* or *observationally equivalent*.

Modeling State and Command Execution

The state of a TPM will minimally be represented as its volatile and non-volatile RAM, PCR values, installed keys, storage root key (SRK) and endorsement identity key (EIK). In the abstract model, a tuple or record structure will be used while the concrete model representation will be derived from the TCG specification.

Defining command execution is defining a function over state that specifies how the command *observes* and *modifies* the state. Observations represent outputs from the TPM while modifications represent the internal state change caused by command execution. Thus, the state of the TPM has type (A, S) where A is the output type and S is the state type.

INDIVIDUAL TPM COMMANDS are modeled as functions over TPM state and input. As an example, PCR extension accepts a PCR identifier, new hash value and a PCR state, and produces a new PCR state and associated output. The PVS function `tpmExtend` is a concrete example of one such function:[2]

[2] `tpmExtend` is taken from a proof-of-concept model developed in PVS.

```
tpmExtend(s:tpmState,n:PCRINDEX,h:HV) : tpmState =
  s WITH ['pcrs := pcrsExtend(pcrs(s),n,h)];
```

The tpmExtend function abstractly defines how an extension request alters the state value associated with a TPM. Specifically, the new state is the old state s with PCR number n extended with hash value h.

TRADITIONAL HARDWARE VERIFICATION USES explicit state passing among functions to model command sequencing. A function is written to perform an operation that accepts a system state and returns a new state resulting from function application exactly as the tpmExtend above. Sequences of operations may now be evaluated by starting with an initial state an explicitly passing the state to each function call. For example, given definitions of pcrExtend and senter functions, an initialization using senter followed by an extension of PCR 5 with the hash value h has the following form:

```
tpmExtend((senter initial),5,h)
```

We propose a less common verification model that uses a *state monad* [Moggi, 1997] to implicitly pass state values rather than a parameter to explicitly pass state. This is a more natural style for modeling where state is an ephemeral data structure updated as an effect of command calls. In a monadic model we use unit, >>=, and >> to initialize and sequence state transformations.[3] unit simply builds an initial state, *s*, on which the command sequence starts by lifting *s* into the state monad:

[3] The operator >>= is called *bind* while >> is called *sequence*.

```
unit(s:S):[A-S] = (LAMBDA (a:A) : (a,s);
```

Given a TPM state, s, return generates a function from an output value, a, to the pair (a,s). In effect, unit constructs an initial state on which to start a sequence of computations. As the output is not known until the computation starts, it is held abstract until the computation begins.

The bind function, f >>= s, takes a state, s, and applies the transformation, f, to it to produce a new state:

```
>>= (m:State,f:[A->State]):State =
   state(LAMBDA(s0:S):
      LET (a,s1) = runState(m)(s0) IN runState(f(a))(s1));
```

The sequence function, f >> s, behaves similarly but does not require the output, a, of a previous bind. Sequences of instructions are modeled by repeated application of the bind and sequence functions, applying the result to an initial state. The sequence used to apply `tpmExtend` to PCR 1, 2 and 3 sequentially to an initial state would have the form:

```
tpmExtend(1,h1)
>> tpmExtend(2,h2)
>> tpmExtend(3,h3)
>>= unit
```

and would be applied to an initial state, `initial`, by running the command sequence:

```
runState(
  tpmExtend(1,h1)
  >> tpmExtend(2,h2)
  >> tpmExtend(3,h3)
  >>= unit)(initial)
```

Processing begins in state `initial` and the function `tpmExtend` is called three times in sequence starting with `tpmExtend(1,h1)`. The state resulting from each function application is threaded through the sequence, obviating the need for an explicit state parameter to artificially pass state.

The infrastructure for running commands is completely reusable allowing execution of arbitrary command sequences. Furthermore, by replacing the state definition, both abstract and concrete representations can be evaluated in this manner.

The real benefit of the monadic approach is the natural appearance of command sequence and the ease of transforming actual TPM command sequences into the formal model. In the example above, command sequencing starts at the top and continues through the bottom as one would expect. We believe that both our abstract and concrete models can live long beyond initial verification and serve as a test bench for those developing new TPM implementations even if they are not versed informal modeling techniques.

IN EARLIER WORK WITH PVS and Isabelle, we have successfully used the state monad for modeling stateful transformation. The data structure and functions defining the monad have been verified and can be reused for this effort.

The Attestation Subset

We informally define the Attestation Subset of the TPM command set as the collection of commands that: (i) manage PCRs; (ii) manage localities; (iii) seal and unseal data; and (iv) generate quotes.

Process Configuration Registers (PCRs) store and extend hashes of BLoBs taken at various times during system boot and execution. PCR values are never directly set, but are instead extended with new hash values. Extending a PCR is defined as replacing an existing hash value with the hash of the concatenation of the existing value with the new hash value. Specifically:

```
pcr' = #(pcr|h)
```

where `pcr` and `pcr'` are the current and next hash value and `h` is the new hash value. At any time the PCR's contents are a measurement representing the trajectory of hash values used to extend it's initial value over time. Pragmatically, this gives us an assessment of both the value and order of hash values.

The TCG [—, 2007] has identified a number of standard PCRs that contain hashes of various elements of the measured launch environment (MLE), sinit and other critical system components.

PCRs are useful for assessing the state of the system whose measurements they represent and *sealing* data or *wrapping* keys. An external appraiser can look at hash values from a system's TPM and assess how it booted without sacrificing confidentiality of the remote system. [Haldar et al., 2004, Goldreich and Oren, 1994] Internally the data sealed to a set of PCRs cannot be unsealed in a state unless its PCRs match those involved in the seals.[4]

[4] We will see the same technique used for wrapping keys.

PCRs are protected by a primitive access control facility known as *locality*. The locality of a PCR indicates what processes can extend its value. One locality is accessible only by hardware while another controls access to the `sinit` measurement. In each case, PCRs associated with a locality are restricted to commands executable in that locality.

WHAT WE REFER TO AS the Attestation Subset are TPM commands and directives that: (i) reset and extend TPM values; (ii) seal and unseal data; and (iii) produce PCR composites for external attestation.

The Key Management Subset

We informally define the Key Management Subset of the TPM command as the collection of commands that: (i) generate symmetric keys, asymmetric key pairs, and nonces; (ii) install and revoke asymmetric keys; and (iii) wrap and unwrap keys as a part of key generation and installation.

A *wrapped* key is an asymmetric key pair whose private key is sealed to a TPM state and whose public key is clear text. When a blob is encrypted with a wrapped key, the public key is used for encryption in the canonical fashion. The private key will be available for decryption only when installed in the TPM associated with the seal with its PCRs in the same state as when the key was wrapped. The key pair is thus associated with a specific TPM in a specific state.

Keys may be chained by using keys to envelope other keys. A key may be wrapped with the private key of an already installed key. Thus, to decrypt any secret encrypted with the key, the key used for wrapping must be installed in addition to standard PCR requirements on the wrapped key.

Common TPM Command Sequences

In addition to individual commands, a number of important command sequences will be verified. These will minimally include measured boot, remote attestation, data migration, and session management. Each of these is described in the following paragraphs.

THE *measured boot sequence* IS INITIATED by a processor call to `senter` and communicates with the TPM to perform operations including PCR extension, unsealing data and installing keys. Specifically, `senter` measures `sinit` into a PCR, executes `sinit` to measure the measured launch environment (MLE), and starts the MLE. At the end of the sequence one can establish that `senter` and `sinit` were called and that a proper MLE was started.

The *attestation sequence* is initiated by an external request for a quote and communicates a signed PCR composite and nonce to the external requester. The requester should be able to establish trustworthiness by assessing the cryptographic properties of the quote and comparing reported hashes with know golden has values.

The *key migration sequence* is initiated when encrypted data is moved from one TPM to another. New keys are generated and existing keys are re-wrapped with new keys for the new TPM. This should occur without exposing encrypted secrets outside the TPM.

The *session management sequence* is initiated when a TPM session is opened and closed. It involves the generation of keys and establishment of key pairs for a session.

We will not limit our work to these activities, but they do describe an initial set of TPM actions to consider. We will work with the sponsor to identify other key TPM tasks for verification.

Statement of Work

The primary outcome of this proposal will be ***a verified instruction-level model of core TPM functionality***. The following outlines basic technical tasks that outline our work plan:

- *Model the TPM state and command execution* — Using monadic techniques develop a model of the TPM internal state and a framework for command execution.

  - Develop an extensible model of TPM state including PCRs, NVRAM, and session management information
  - Define a standard model of TPM command execution over TPM state

- *Model and Verify the attestation subset* — Using the TPM state model, specify and verify the TPM command subset for PCR management and quote generation. Extend existing verification of PCR extension results.
  - Identify the core command subset associated with PCR management and quote generation
  - Define a denotation of the attestation subset to the monadic command execution model
  - Define requirements for the attestation subset
  - Verify attestation subset against specified requirements

- *Model and Verify the key management subset* — Using the TPM state model, specify and verify the TPM command subset for generating and managing keys.
  - Identify the core command set associated with key generation, wrapping and installation.
  - Define a denotation of key management commands to the monadic command execution model
  - Define requirements for the key management subset
  - Verify key management subset against specified requirements

- *Verify common TPM command sequences* — Use the new monadic model to expand previously performed verification of various TPM command sequences.
  - Define trust properties associated with TPM-based provisioning, boot, run-time, and tear-down processes that will minimally include:
    * Confidentiality of TPM data and protected secrets
    * Monotonicity of PCR extension
    * Integrity of PCR data and quotes
    * Availability of PCR data and quotes
    * Verifiable quote authenticity
  - Identify TPM command sequences associated with common activities that will minimally include:
    * Provisioning and ownership
    * Opening and closing sessions

  - PCR management including reset, extension, and locality
  - Seal/unseal and wrap/unwrap functionality
  - Key generation, installation, and removal
  - Generating and interpreting simple and complex TPM quotes
  - Data migration among TPM instances
- Verify that trust properties are invariant over TPM command sequences

UNLESS DIRECTED OTHERWISE, verification will be performed using PVS, SAL, and yices. It is anticipated that the vast majority of the work will be done using PVS, but we will explore defining data structures for TPM state that can be used in both PVS and SAL.

Analysis and Summary

WHAT IS THE PROBLEM, WHY IS IT HARD? If the TPM is to form the trusted root of secure computing systems, validation, verification, and certification are essential tasks. Without assurance in the root-of-trust, there is no trust in the system built from it. With millions of TPMs fielded, it becomes critical to: (i) understand what we have specified; (ii) understand what we have implemented; and (iii) assure our implementations are consistent with what we have specified.

The currently fielded TPM is based on the version 1.2 definition. This specification is an English text requirements document that is known to suffer from inconsistency and omission. The only mechanisms available for certification are compliance test suites made available by the TCG. As such, the TPM specification is difficult to understand, implement, and certify. [5]

[5] This is not to blame the TCG – this the current state-of-the-practice for such systems.

Given this, it is not surprising that no known manufactured hardware device is 100% compliant with the entire TPM specification. Among the best and most compliant implementations is the Berlios emulator [Strasser and Stamer, 2008] used by numerous developers for development and testing. As a software implementation, it is not viable as a core root-of-trust for most systems.

JASON grand challenge

HOW IS IT SOLVED TODAY? Verification and certification of current TPM implementations is done using current best practices that involve compliance testing for certification. Conformance test suites are available from the TCG, IBM and others to demonstrate correctness of a TPM implementation with respect to the TCG specification. Specifically, the TCG's compliance test suite is used for certification. A TPM is considered correct if it executes all test cases successfully.

A TPM implementation is certified by the TCG separately using the same test suite as defined by their web site.

The TPM version 2.0 specification is due from the TCG in 2012. This specification is held proprietary by the TCG membership, however it is known that the specification will be executable. This should make verification simpler and will allow testing of the specification. However, it will not enable a substantially more robust verification process and cannot guarantee full device specification coverage.

WHAT IS THE NEW TECHNICAL IDEA; WHY CAN WE SUCCEED NOW? The technical innovation is application of established formal methods techniques to the TPM specification. We believe this is achievable based on past experience with formal verification generally and with TPM related verification specifically.

We have used formal verification tools – PVS and SAL specifically – to model various aspects of Intel's TXT system boot using a TPM. To achieve this, we developed a highly abstract TPM model that emulates selected TPM functionality. Thus, definitions exist for TPM functionality as well as TPM-base measured boot.

We and others have used monadic state threading models [6] to define stateful systems in the PVS and Isabelle [Nipkow et al., 2002] verification tools. The monadic model simplifies handling state transitions. Such transitions are prevalent in processor modeling generally and TPM processor modeling specifically. The state monad achieves handling state without making state an explicit parameter in models.

6

We have developed and demonstrated a prototype TPM model using monadic techniques in PVS. The current model implements elements of the PCR, remote attestation, and key management subsystems of interest in this effort. These preliminary models are far from complete, but successfully demonstrate the approach and that specifications written using these techniques are accessible to those with minimal formal methods training.

We have developed highly abstract models of TPM processes used for verification of system boot, access control, and measurement techniques. These do not represent the internals of the TPM, but use a TPM model in larger trusted computing activities. What we have learned in these studies supports the contention that formal modeling is feasible and effective. Further, knowledge of TPM use will be leveraged in this effort.

WHAT IS THE IMPACT IF SUCCESSFUL? There are three outcomes of this process that will have value to the community: (i) a formal TPM specification; (ii) verification of the formal specification; and

(iii) a verified TPM implementation. A formal TPM specification will remove ambiguities and inconsistencies in the current TPM specification. Verifying the formal specification will assure its correctness and guide developers in exploring the current specification. A verified TPM – the ultimate research goal – will provide an implementation that is consistent with the verified specification, dramatically simplifying certification.

HOW WILL THE PROGRAM BE ORGANIZED? The technical program lead will be Dr. Perry Alexander with the vast majority of the technical effort performed by PhD seeking graduate research assistants. Models and verification results will be maintained in a git repository accessible to the sponsor and with the sponsors approval, accessible in the public domain.

HOW WILL INTERMEDIATE RESULTS BE GENERATED? Intermediate results will consist of individual specifications and their associated verification. In particular, the state model is the first important result followed by instruction specification and verification. Each instruction can be modeled in a relatively orthogonal fashion whose results can be used and evaluated immediately.

HOW WILL YOU MEASURE PROGRESS? The structure of the program around three major modeling tasks: (i) state modeling; (ii) instruction-level modeling of PCR management, key management, and quote generation; and (iii) protocol modeling. Progress can easily be measured by looking at successful completion of state modeling followed by completion of individual instructions and instruction sets. Protocol modeling will occur concurrently with modeling instructions to inform the modeling task and validate modeling results.

WHAT WILL IT COST? Ideally we would use 2 PhD students on this effort for two years. See accompanying budget for detailed costs.

More Information

For more information on the contents of this proposal and research at ITTC, please contact:

Dr. Perry Alexander
Director, Information and Telecommunication Technology Center
Professor, Electrical Engineering and Computer Science
The University of Kansas
+1.785.864.7741
`alex@ittc.ku.edu`

For more information on ITTC, please visit:
`http://www.ittc.ku.edu`

For more information on establishing nondisclosure agreements and confidential data agreements, please contact:

Mr. Keith Braman
Director, ITTC Technology Commercialization
+1.785.864.7697
kbraman@ittc.ku.edu

References

—. *TCG TPM Specification*. Trusted Computing Group, 3885 SW 153rd Drive, Beaverton, OR 97006, version 1.2 revision 103 edition, July 2007. URL https://www.trustedcomputinggroup.org/resources/tpm_main_specification/.

S. Bensalem, V. Ganesh, Y. Lakhnech, C. Munoz, S. Owre, H. Rueb, J. Rushby, V. Rusu, H. Saidi, N. Shankar, E. Singerman, and A. Tiwari. An overview of SAL. In C. M. Holloway, editor, *Fifth NASA Langley Formal Methods Workshop*, Williamsburg, VA, June 2000.

B. Dutertre and L. de Moura. A Fast Linear-Arithmetic Solver for DPLL(T). In *Proceedings of The International Conference on Computer Aided Verification (CAV'06)*, 2006.

O. Goldreich and Y. Oren. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7:1–32, 1994. ISSN 0933-2790. URL http://dx.doi.org/10.1007/BF00195207. 10.1007/BF00195207.

V. Haldar, D. Chandra, and M. Franz. Semantic remote attestation – a virtual machine directed approach to trusted computing. In *Proceedings of the Third Virtual Machine Research and Technology Symposium*, San Jose, CA, May 2004.

E. Moggi. Metalanguages and applications. In A. M. Pitts and P. Dybjer, editors, *Semantics and Logics of Computation*, volume 14, pages 185–239. Cambridge University Press, Cambridge, 1997. URL citeseer.nj.nec.com/moggi95metalanguages.html.

T. Nipkow, L. C. Paulson, and M. Wenzel. *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*, volume 2283 of *LNCS*. Springer, 2002.

S. Owre, J. Rushby, and N. Shankar. PVS: A Prototype Verification System. In D. Kapur, editor, *Proc. of 11th International Conference on*

Automated Deduction, volume 607 of *Lecture Notes in Artificial Intelligence*, pages 748–752, Saratoga, NY, June 1992. Springer–Verlag.

D. Sangiorgi. *Introduction to Bisimulation and Coinduction*. Cambridge University Press, 2012.

M. Strasser and H. Stamer. A software-based trusted platform module emulator. In P. Lipp, A.-R. Sadeghi, and K.-M. Koch, editors, *Trusted Computing - Challenges and Applications*, volume 4968 of *Lecture Notes in Computer Science*, pages 33–47. Springer Berlin / Heidelberg, 2008. ISBN 978-3-540-68978-2.

PROPOSED BUDGET

Period 1: 05/01/12 - 09/30/12

SALARIES AND WAGES							
Senior Personnel		% time	Months	Rate			
Perry Alexander, PI							
summer (0.25 months)		8.334	3.0	13,134	3,284		
Total senior personnel						3,284	
Other Personnel	Persons	% time	Months	Rate			
Graduate Student(s)							
academic 05/01/12 - 05/15/12	2	50	0.5	3,441	1,721		
summer 05/16/12 - 08/15/12	2	100	3.0	3,441	20,646		
academic 08/16/12 - 09/30/12	2	50	1.5	3,441	5,162		
Total other personnel						27,529	
Total salaries and wages						30,813	
FRINGE BENEFITS							
35% faculty and staff					1,149		
15% students (employed 76% or more)					3,097		
7% students (employed 75% or less)					482		
Total fringe benefits						4,728	
Total salaries, wages & fringe benefits							35,541
EQUIPMENT							
Total equipment							0
TRAVEL							
(a) to Battelle - Columbus, Ohio							
	# Persons	Trips	Days	Amount			
	2	2	4				
Transportation (airfare)				473.55	1,894		
Per diem				54	864		
Lodging				160	2,560		
Car rental				50	400		
Total (a)						5,718	
Total travel							5,718
OTHER DIRECT COSTS							
Research materials & supplies						1,000	
Publications (copying and distribution of research results)						0	
Consultant Services						0	
Computer Services						0	
Subawards						0	
Tuition	Su. 12	Fall 12					
2 GRA	1206	3835			10,082		
Communications (long distance, fax, postage)					0		
Computer networking and maintenance costs					2,958		
Total Other Direct Costs							14,040
TOTAL DIRECT COSTS							55,299
BASE							45,217
INDIRECT COSTS On-Campus Research (47% of total direct costs excluding equipment and tuition allowance)							21,252
TOTAL PROPOSED COSTS - Period 1							**$76,551**

PROPOSED BUDGET (Continued)

Period 2: 10/01/12 - 09/30/13

SALARIES AND WAGES

Senior Personnel		% time	Months	Rate			
Perry Alexander, PI							
summer (0.25 months)		8.334	3.0	13,791	3,448		
Total senior personnel						3,448	
Other Personnel	**Persons**	**% time**	**Months**	**Rate**			
Graduate Student(s)							
academic 10/01/12 - 04/30/13	2	50	7.0	3,441	24,087		
academic 05/01/13 - 05/15/13	2	50	0.5	3,630	1,815		
summer 05/16/13 - 08/16/13	2	100	3.0	3,630	21,780		
academic 08/15/12 - 09/30/13	2	50	1.5	3,630	5,445		
Total other personnel						53,127	
Total salaries and wages						56,575	

FRINGE BENEFITS

35% faculty and staff	1,207		
15% students (employed 76% or more)	3,267		
7% students (employed 75% or less)	2,194		
Total fringe benefits		6,668	
Total salaries, wages & fringe benefits			63,243

EQUIPMENT

Total equipment	0

TRAVEL

(a) to Battelle - Columbus, Ohio

	# Persons	Trips	Days	Amount			
Transportation (airfare)				500	0		
Registration				350	0		
Per diem				54	0		
Lodging				160	0		
Car rental				50	0		
Total (a)						0	
Total travel							0

OTHER DIRECT COSTS

	Spring 13	Sum 13	Fall 13			
Research materials & supplies					1,000	
Publications (copying and distribution of research results)					0	
Consultant Services					0	
Computer Services					0	
Subawards					0	
Tuition 2 GRA	3835	1269	4045	18,298		
Communications (long distance, fax, postage)				0		
Computer networking and maintenance costs				4,497		
Total Other Direct Costs						23,795

TOTAL DIRECT COSTS	87,038
BASE	68,740
INDIRECT COSTS On-Campus Research (47% of total direct costs excluding equipment and tuition allowance)	32,308
TOTAL PROPOSED COSTS - Period 2	**$119,346**

PROPOSED BUDGET (Continued)

Period 3: 10/01/13 - 09/30/14

	Persons	% time	Months	Rate			
SALARIES AND WAGES							
Senior Personnel		% time	Months	Rate			
Perry Alexander, PI							
summer					0		
Total senior personnel						0	
Other Personnel	Persons	% time	Months	Rate			
Graduate Student(s)							
academic 10/01/13 - 04/30/14	2	50	7.0	3,630	25,410		
Total other personnel						25,410	
Total salaries and wages						25,410	
FRINGE BENEFITS							
35% faculty and staff					0		
15% students (employed 76% or more)					0		
7% students (employed 75% or less)					1,779		
Total fringe benefits						1,779	
Total salaries, wages & fringe benefits							27,189
EQUIPMENT							
Total equipment							0
TRAVEL							
Total travel							0
OTHER DIRECT COSTS							
Research materials & supplies						0	
Publications (copying and distribution of research results)						0	
Consultant Services						0	
Computer Services						0	
Subawards						0	
Other:							
Tuition	Spring 14						
2 GRA	4045				8,090		
Communications (long distance, fax, postage)					0		
Computer networking and maintenance costs					1,903		
Total "Other"						9,993	
Total Other Direct Costs							9,993
TOTAL DIRECT COSTS							37,182
BASE							29,092
INDIRECT COSTS (47% of total direct costs excluding equipment and tuition allowance)							13,672
TOTAL PROPOSED COSTS - Period 3							**$50,854**
TOTAL PROPOSED COSTS - All Periods							**$246,751**