U.S. DEPARTMENT OF
ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

□ð

SCI┐FI□: Supply □ð Chain Integration □ð For □ð Inte

□ð

7/2/2012

Pacific Northwest
NATIONAL LABORATORY

Proudly Operated by Battelle *Since 1965*

CONTENTS

FIGURES

ACRONYMS & ABBREVIATIONS

ALU	Arithmetic Logic Unit
AMD	Advanced Micro Devices, Inc.
ARM	Advanced RISC Machines
ASCR	Advanced Scientific Computing Research
AST	Abstract Syntax Tree
BSD	Berkeley Software Distribution
CCEV	Common Criteria Evaluation and Validation
CEDS	Cybersecurity for Energy Delivery Systems
CERT	Computer Emergency Response Team
CPD	Cyber-Physical Device
CPU	Central Processing Unit
CRTM	Core Root of Trust Measurement
DMA	Direct Memory Action
DMI	Digital Management, Inc.
DoD	Department of Defense
DOE	Department of Energy
D-RTM	Dynamic Root of Trust for Measurement
DSM	Data Storage and Management
EDG	Edison Design Group
EDS	Electric Delivery System
EMS	Energy Management System
EPRI	Electric Power Research Institute
ESD	Electrostatic Discharge
FOL	First Order Logic
FPGA	Field-Programmable Gate Array
FSM	Finite State Machine
FY	Fiscal Year
HPC	High Performance Computing
I/O	Input/Output
IC	Integrated Circuit
IED	Intelligent Electronic Device
IF-M	Interface-map
IMM	Integrity Management Model
IP	Intellectual Property
IR	Intermediate Representation
ISO	International Organization for Standardization
IT	Information Technology
LLNL	Lawrence Livermore National Laboratory
LLVM	Low Level Virtual Machine
MHz	Megahertz
MPI	Message Passing Interface

NIST	National Institute of Standards and Technology
NSA	National Security Administration
ORNL	Oak Ridge National Laboratory
OS	Operating System
PC	Personal Computer
PCI	Peripheral Component Interconnect
PCR	Platform Configuration Registers
PG&E	Pacific Gas and Electric
PNNL	Pacific Northwest National Laboratory
POC	Point of Contact
PTS	Platform Trust Services
R&D	Research & Development
RE	Reverse Engineering
RoT	Root of Trust
RTM	Root of Trust for Measurement
RTU	Remote Terminal Unit
SAS	Substation Automation System
SAT	Satisfiability
SCADA	Supervisory Control and Data Acquisition
SCI-FI	Supply Chain Integration for Integrity
SEI	Software Engineering Institute
SMT	Satisfiability Modulo Theories
SoC	System on Chip
S-RTM	Static Root of Trust for Measurement
T&D	Transmission & Distribution
TA	Trust Anchor
TC	Technical Community
TC	Trusted Computing
TCB	Trusted Computing Base
TCG	Trusted Computing Group
TM	Trust Model
TNC	Trusted Network Connect
TPM	Trusted Platform Module
TRoT	Transitive Root of Trust
UPC	Unified Parallel C
WAMP&C	Wide Area Monitoring Protection & Control
XML	eXtensible Markup Language

EXTENDED FIELD WORK PROPOSAL

Project Area 3

Supply Chain Integration for Integrity

For
Office of Electricity Delivery and Energy
Reliability Research and Development Division
In Response To
RC-CEDS-2012-02

Program Manager: Paul Skare

Technical POC: David Manz

1 Project Summary

Lead Laboratory: Pacific Northwest National Laboratory

Other Team Members: Oak Ridge National Laboratory, Lawrence Livermore National Laboratory, Pacific Gas and Electric, Digital Management, Inc.

Project Director: David Manz, Ph.D. (PNNL)

Co-Investigator(s): Josef Allen, Ph.D. (ORNL), Dan Quinlan, Ph.D. (LLNL), Ken Masica (LLNL), Jessica Smith (PNNL)

Project Title: Supply Chain Integration For Integrity

Project Objectives: An interdisciplinary team of researchers and developers from across the DOE complex and industry propose to address critical supply chain challenges. At this time there are no ISO standards for electricity transmission and distribution (T&D) systems for supply side security. Commercially, a large percentage of these systems are manufactured overseas. Hence, our supply chain is hard to manage until after a product is purchased. What we propose is rather than simply inspecting products, we seek to create a total quality management system where we build supply chain integrity into the systems that are procured. If we do not take the approach of built -in supply chain integrity then this will introduce critical concerns about the reliability of any system using those devices and poses a threat to national security. No longer can we just inspect the manufactured product for performance, longevity, etc. We must now inspect for malicious intent. This requires a subcomponent level review. Subcomponents include integrated circuits (ICs), memory, firmware, printed circuit board s, PCI slot devices such as network interface cards, and graphics card s.

Most of the ICs associated software that run our computers, cell phones, nuclear power plants, and national energy delivery infrastructure are produced overseas. We have few practical methods of determining that the approved design is actually implemented in the ICs, firmware, and software. Our lack of awareness introduces critical concerns about the reliability of any system using those devices and could be obscuring a threat to national security. Current methods to reverse engineer ICs, firmware, and software are too expensive or destructive to be practically usable for most users. Easily distributable, cost effective, and domain-relevant methods are needed to reverse engineer the state machines that form the logic of these critical devices in the hardware, firmware, and software.

Project Description: We propose to address the challenges of supply chainin an integrated manner. The project is divided into three prongs: the first addresses policy and architecture for built-in supply chain integrity of trusted components; the second analyzes software and firmware; and the third evaluates hardware supply chain concerns. The proposed integrated approach allows the team to leverage specific capabilities and each partner's areas of expertise, while still ensuring the broader issues of energy delivery supply chain integrity are addressed.

For supply chain policy and architecture, tools and techniques will be developed to address two types of supply chain architecture integrity: static and dynamic discovery. Static supply chain refers to discovering a compromise of digital assets after manufacturing but before commissioning into service. Dynamic discovery refers to detecting compromise of digital assets for the electric delivery system (EDS) during their period of service. The dynamic and static discoveries hinge on the trusted computing base (TCB). In its totality, the TCB is the heart of a trusted computer system. The TCB contains all of the elements of the system responsible for supporting the security policy and supporting the isolation of objects (code and data) on which the protection is based. Furthermore, we will study supply chain as a process to better understand when tools and techniques, current and to be developed, will be best applied in context with the industry and global nature of commerce.

For supply chain firmware and application software, we will develop the analysis capabilities to support custom forms of analysis for different types of embedded software used in power automation field devices as well as application software used for power system monitoring and control. Analysis capabilities will span both source code and binary executables and leverage the existing ROSE software analysis framework developed at LLNL. LLNL will extend the ROSE framework, as required, to support new analysis and expected processors or micro-controllers specific to our supply chain vendor partners. New and existing ROSE analysis tools will be used to support the evaluation of both patches (incremental changes to existing software) and full applications.

For supply chain hardware, our solution performs an intelligent brute force exploration using the available input/output (I/O) to determine all possible states to a pre-specified depth; identifies loops, terminators, and equivalent states; and then reduces the state set to a minimal set. This minimal set is a very close equivalent to the actual state machine the designers used to create the IC. If a design is known, this reverse engineered state machine can be compared to the original to ensure nothing was inserted during the manufacturing of the IC; if no design is known, we can begin to work upwards, improving the understanding of the functionality of the IC, to determine that it does not contain unwanted or malicious logic. This technique allows us to have confidence in the small to medium sized ICs that are the building blocks of our world.

Impact: A suite of open source tools and technology will be available to address the needs of supply chain integrity for end utilities, vendors, and chipset manufacturers. The suite will range from stand -alone tools that can be run locally to provide hardware supply chain assurances, to large scale high performance computing services that statistically analyze systems of systems to identify potential concerns in critical infrastructure supply chains. Our work will also be relevant broadly to other industries as part of support for more general supply chain integrity. We will demonstrate how to use these tools as part ofthe validation of our research work with our vendor partners to obtain feedback to our iterative development process with the electric power industry throughout the project's lifetime.

2 Technical Content

2.1 Introduction

On January 25, 2012 President Obama made the announcement for the National Strategy for Global Supply Chain Security. This strategy focuses on the worldwide network of transportation, postal, shipping assets, and supporting infrastructures, and articulates our national vision and approach, and encourages collaborative implementation with key state, local, tribal, territorial, private sector , and international stakeholders. This strategy focuses on two primary goals:

1) Promote the Secure and Efficient Movement of Goods,
2) Foster a Resilient Supply Chain.

The first strategic thrust, "Promoting the Secure and Efficient Movement Goods," has several core foci:

- Resolve threats early, meaning integrate the security process into the supply chain operations
- Improve verification and detection capabilities of those transported and used goods
- Enhance security of infrastructures and conveyances by t reating the supply chains import and export areas with a higher level of national security scrutiny
- Maximize the flow of legitimate trade through making it economically and procedurally easier

The second strategic thrust, "Foster a Resilient Supply Chain," also has multiple foci:

- Mitigate systematic vulnerabilities by using risk management policies to identify and protect key critical infrastructures and their ancillary components
- Promote trade resumption policies and practices for global restoration by im plementing national and global paradigms for co-ordination of the movement of goods after a disruption

Electric power grid operation involves equipment and systems that utilize software components that range from embedded processor code in automation equipment in generation plants, transmission substations, and smart power meters, to SCADA software running on commercial servers and operating systems that collect, process, and display large amounts of operational data. The ability to analyze embedded code, firmware, hardware, and application software for potential security vulnerabilities can improve the electric power industry supply chain integrity and provide an increased degree of assurance in product selection and implementation. For example, automated code analysis tools and techniques have the potential to determine differences in binary and source code product versions, find potential backdoors and other security risks, and map inputs and outputs to identify unintended execution paths through the code. Highly tailored forms of software analysis tools are needed to work on source code/binaries and hardware produced by electric power industry vendors and installed by customers such as the utilities at the end of the supply chain. *Supply Chain Integration for Integrity* directly addresses the pressing needs articulated in the announcement for the National Strategy for Global Supply Chain Security.

2.2 Partnerships

A diverse team of researchers and developers from across the DOE complex and industry propose to address critical supply chain challenges.

2.2.1 ORNL

Oak Ridge National Laboratory (ORNL) is the largest science and energy national laboratory in the Department of Energy (DOE) system. ORNL's scientific programs focus on materials, neutron science, energy, high-performance computing, systems biology, and national security.

ORNL partners with the state of Tennessee, universities, and industries to solve challenges in energy, advanced materials, manufacturing, security, and physics. The laboratory's science and technology innovations are translated into applications that add value throughout the world.

2.2.2 PNNL

Pacific Northwest National Laboratory's (PNNL's) mission is to deliver leadership and advancements in science, energy, national security, and the environment for the benefit ofDOE and the nation. PNNL is fully committed and capable of supporting the DOE mission for Energy Delivery Systems Research. PNNL staff provide unique hardware analysis and reverse engineering capabilities that can be applied to critical infrastructure and energy delivery systems.

2.2.3 LLNL

Lawrence Livermore National Laboratory (LLNL) is a DOE national security laboratory focused on a broad range of research work including that associated with national security, high performance computing (HPC), nuclear physics, and electrical engineering . LLNL is committed to supporting the cyber security for DOE mission for energy delivery systems research. LLNL has unique capabilities focused on both source code and binary analysis required for software assurance and cyber security, supporting the development of tools to automate the specialized analysis requirements. This work will be used to target supply chain integrity for the electric power industry.

2.2.4 Pacific Gas and Electric

Pacific Gas and Electric Company, incorporated in California in 1905, is one of the largest combination natural gas and electric utilities in the United States. Based in San Francisco, the company is a subsidiary of PG&E Corporation.

There are approximately 20,000 employees who carry out Pacific Gas and Electric Company's primary business—the transmission and delivery of energy. The company provides natural gas and electric service to approximately 15 million people throughout a 70,000-square-mile service area in northern and central California.

PG&E has agreed to provide consulting and expertise on tool design, development, and evaluation in support of the *Supply Chain Integration for Integrity* project in an ongoing and as needed basis. Additionally, PG&E will work with PNNL and LLNL to devise appropriate and realistic testing and evaluation environments.

2.2.5 DMI

Digital Management, Inc. (DMI) is a leading IT solutions and business strategy consulting firm focused on providing solutions that transform enterprise operations in government and business by dependably bridging the gap between business strategy and mission success. Leveraging technology as an efficient, economical means to an end, DMI crafts solutions that result in increasingly interoperable, responsive, and cost-effective enterprises. DMI's commitment to superior services and solutions, including strategic business transformation, trusted computing, supply chain improvement, software systems modernization, enterprise information management, cyber security, and healthcare IT, has resulted in dramatic revenue growth, and a growing client base that now includes fourteen of the fifteen U.S. Federal Departments. DMI is headquartered in Bethesda, Maryland with satellite and project offices throughout the world.

DMI brings specific expertise in creating industry-driven security solutions to the Department of Defense and federal civilian agencies, with a focus on supply chain security and efficiency and on trusted computing solutions.

2.3 Built-in Supply Chain Integrity Architecture and Policy

The approach of this project is to identify areas of significant vulnerability and a threat to our national security posture. One of those areas that were found to be at significant risk was the supply for our electric delivery system (EDS). It is estimated that 90 percent of all manufactured assets forthe EDS are imported. A large percentage of those assets have digital components. Whether batteries, transformers, etcetera, it is found that most have some form of computational derivative associated with them and are not made in the United States. Typically, EDS devices have a cyber-physical part to them. We can define cyber-physical devices (CPDs) as those devices that turn an analog signal to a digital signal, and based on that digitized input, makes a decision such that some transduction takes place to alleviate an issue. An example is an intelligent electronic device (IED). A typical CPD has a microprocessor (central processing unit (CPU)) that runs some sort of operating system (OS) with mostly deterministic applications used for control, communication, commands, and signals. The processors for CPDs are quite diverse and the typical CPD CPU includes the need for extended temperature ranges (-40/+85'C), low power with fan-less operation, low cost, and small footprint, etc. A typical variety of CPU architectures (and their vendor(s)) include integrated architecture (Intel/AMD), power architecture (PowerPC), and advanced risk machine (many vendors). Along with this is a diverse mix of operating systems (Windows, Linux, and Real Time OS (RTOS) various implementations). We mention the fact that CPD have CPU, OS, and applications to make an observation. That observation is that most CPU and OS have a trusted computing component. Why? The Department of Defense (DoD) market has primarily driven this. Since 2007, DoD mandated that allserver, PC, and laptops built for their consumption must have a trusted computing component; this included the Intelligence Community. As the DoD is one of the largest consumers in the world, it created a paradigm shift throughout the world of manufacturing as it pertains to CPU and OS. Hence, most CPUs and OSs support some form of trusted computing (TC) architecture. We will seek to leverage the TC architecture to provide enterprise supply side security for the electric delivery system.

Trusted computing is centered on the concept of root of trust (RoT). The RoT is defined as a component that must always behave in the expected manner (immutable), because its misbehavior cannot be detected. The RoT is what helps to form the Trusted Computing Base (TCB). The TCB complete set of roots of trust has at least the minimum set of functions to enable a description of the platform characteristics that affect the trustworthiness of the platform. These platform characteristics are centered on the coreroot of trust measurement (CRTM). Hence, we sum this up as the trust anchor (TA). The Trusted Computing Group (TCG) uses the well-known Trusted Platform Module (TPM) as their standardized TA. TCG is a not-for-profit organization formed to develop, define, and promote open, vendor-neutral, industry standards for trusted computing building blocks and software interfaces across multiple platforms.

However, since TPM is not the only trusted anchor that can be leveraged, we will use the most generic term TA. With respect to most trusted anchors there is the notion of derived transitive trust that emanates from the RoT. For the CEDS call we will use RoT and the notion of transitive trust to propose a solution for detecting compromise of supply chain integrity. This is project 3 ofthe RC-CEDS-2012-02.

There are two types of supply chain integrity that we will address: static and dynamic discovery. Static supply chain discovery refers to the discovery of digital asset compromise after manufacturing but before commissioning into service. Dynamic discovery refers to the detecting compromise of digital assets for the EDS during their period of service. The dynamic and static discoveries hinge on the TCB. In its totality, the TCB is the fundamental concept of a trusted computer system. It is the portion of the system that is relied on to enforce the security policy of the platform. The TCB "contains all of the elements of the system responsible for supporting the security policy and supporting the isolation of objects (code and data) on which the protection is based. The bounds of the TCB equate to the "security perimeter" referenced in some computer security literature. In the interest of understandable and maintainable protection, a TCB should be as simple as possible consistent with the functions it has to perform. Thus, the TCB includes hardware, firmware, and software critical to protection and must be designed and implemented such that system elements excluded from it need not be trusted to maintain protection. Identification of the interface and elements of the TCB along with their correct functionality therefore forms the basis for evaluation. With respect to TCB is the root of trust management (RTM). We propose to use the RTM as a basis for static and dynamic supply chain security. Two goals are to be able to prove that the TCB was properly established and to allow access to sealed data only if the TCB was properly established.

For the static root of trust for measurement (S-RTM), measurements are taken during the boot process with the intent of being able to demonstrate, through platform configuration registers (PCRs), what software was run during the boot process. This reflects the fact that every software component in the boot path has the potential to modify the TCB of the loaded operating system. A property of the S-RTM sequence is that the evaluation of the TCB of an OS has a dependency on software/firmware that has no ongoing role in the operation of the platform. Software that is loaded, run, and discarded during the boot process must be evaluated to determine if it could or did compromise the TCB.

The purpose of the dynamic root of trust for measurement (D-RTM) is to reduce the complexity of the TCB so that evaluation of the platform state becomes more tractable.

The method of D-RTM is relatively simple. It is observed that a new chain of trust can be started if

1) all processors are placed in a known state,
2) a processor begins running measured code, and
3) the code that the processor has measured cannot be modified except as directed by the measured code running on the processor.

These conditions are met at system reset when the CPU and the rest of the chipset are forced to an initial condition by a hardware signal; system reset is the way in which the S-RTM is started. For D-RTM, the same three conditions are met but without requiring that the full system be reset.

New instructions have been added to processors that cause the CPU to extend the hash of a block of code to the trusted platform module (TPM). Before or during the measurement of this code, it is protected from direct memory action (DMA) access either by locking it in the CPU cache or by extending DMA protections over the code in memory. Protecting the code from DMA ensures that it can't be modified after it is measured and before it is executed. The processor then begins execution of the measured code, starting the D-RTM chain of trust.

An essential difference between the D-RTM chain and the S-RTM chain is that the D-RTM chain can start with the platform hardware already configured and with memory pre-populated with an OS. Thus, it is not necessary to re-run all of the software or firmware that was used to configure the hardware or load the OS. If the code is not run then it is excluded from the TCB as long as any possible effect of that code on the TCB can be negated.

The way in which the D-RTM process neutralizes any possible impact of the boot code is to check to see that the hardware configuration is in a TCB-safe state. That is, it must check to see that the hardware is set up in such a way that the TCB can protect itself. This check is possible because the CPU command that started the D-RTM chain has protected the code doing this check so that no hardware settings can subvert the checks.

We propose to study S-RTM and D-RTM for CPDs This is a new concept and will require us to identify TCB for CPD.

2.4 Application Software and Firmware Supply Chain Analysis

LLNL will develop the analysis capabilities to support custom forms of analysis for both embedded field device firmware and energy management system application software used in the monitoring and control of the electric grid. Analysis capabilities will span both source code and binary executables using the existing ROSE infrastructure at LLNL and extending it, as required, to support analysis specific to the integrity of the electric power industry supply chain. Existing static and symbolic analysis tools will be used to support the evaluation of security patches, firmware updates, and ap plication software. We will define open source tools that can be made available to both vendors and customers.

Our team has unique, combined experience with software assurance technologies, including static and dynamic analysis, source and binary analysis, and formal methods that can be applied to the supply chain integrity challenge addressed in this proposal. LLNL has extensive expertise in source code and binary analysis tools through the ROSE project. LLNL also has large-scale parallel supercomputing resources to support previously demonstrated research on how to apply greater computing intensity for deeper forms of analysis or on larger scale software. However, tools developed for practical use in this proposal will mostly target more modest desktop computer resources expected to be available at vendors.

The open source ROSE compiler infrastructure, developed at LLNL, will be used as our foundation. ROSE is already in used worldwide, supporting research and development of source code and binary analysis tools for a wide range of research areas including cyber security, software assurance, and DOE Exascale research. Via its uniform, extensible internal representation for both source code and binaries, ROSE also helps correlate analysis results generated by different tools. This allows for tool results from non-ROSE-based tools to be assembled into a larger analysis framework, greatly expanding the scale at which ROSE can be used to perform supply chain analysis by integrating third-party tools.

We will conduct the proposed work over three years (see Management Plan), including research, development, and demonstration of the tools. The proposed work will result in new and enhanced tools packaged within existing tool frameworks (e.g., Compass for static analysis tools) built using the ROSE open source compiler infrastructure; plus separate tools to be built using ROSE for specialized forms of analysis specific to capabilities developed within our research. Our proposed work has the advantage of leveraging significant work developed within ROSE and research supported directly by the DOE for over a decade.

An example of a tool built using ROSE is the Compass Analysis Tool which supports analysis of both source code and binaries for violations of software assurance rules. Assurance rules can be either user-defined or taken from existing software assurance related standards. Checkers for rules in Compass are implemented as dynamic plug-ins that interface to the core ROSE infrastructure and provide an abstraction layer over complex analysis algorithms to simplify the act of writing custom plug-ins by tool developers. Currently, Compass includes 150+ checkers to find style, security, correctness, and portability vulnerabilities in both source code and binary executables. It defines a full open source package supporting software analysis, both in source and binary form. It is anticipated that a tool similar to Compass will be developed as part of this proposal to address software assurance issues that are of concern to both vendors and customers in the electric power industry. For example, utilities at the end of the supply chain desire enhanced assurance that new firmware releases for automation equipment in their substation contain only code with the intended functionality and not code with undesired functionality that impact reliability.

2.4.1 Technical Approach

Our work will apply formal methods to create techniques for specifying and verifying software and hardware systems and then leverage the ROSE analysis infrastructure to develop and demonstrate tools that support software analysis across source code and binaries within the electric power industry supply chain.

2.4.1.1 The ROSE Infrastructure

ROSE, developed at LLNL, is an award-winning, open source compiler infrastructure to build program transformation and analysis tools for large scale sequential (C, C++, Java, Fortran) and parallel (OpenMP, Message Passing Interface (MPI) and Unified Parallel C (UPC)) applications. As shown in Figure 1, it can process both source code and binary executables (x86, PowerPC, ARM instruction sets) as input with help from source parsers (e.g., EDG [20]) and disassemblers (ROSE supports its own disassemble technology, which leverages formal methods and symbolic analysis, and can also use other external disassembly tools such as IDA Pro [22]). A uniform abstract syntax tree (AST) is generated as its intermediate representation (IR). Sophisticated compiler analyses, such as control flow and dependence analyses, are developed on top of the AST and encapsulated as simple function calls, which can be readily reused by tool developers. The intended users of ROSE could be either experienced compiler researchers or library and tool developers who may have minimal compiler experience. For binary analysis tool development, some more specialized expertise can be required.

ROSE is particularly well suited for building custom tools for static analysis, program optimization, arbitrary program transformation, domain-specific optimizations, complex loop optimizations, performance analysis, and cyber security. ROSE is released under a BSD-style license. Due to ROSE's significant contributions to software analysis and worldwide adoption into the research and development communities, the ROSE team was awarded the prestigious R&D 100 Award in 2009. This proposal would build on or add specific forms of formal methods capabilities to ROSE in order to build new tools and support a wider range of research within software assurance.

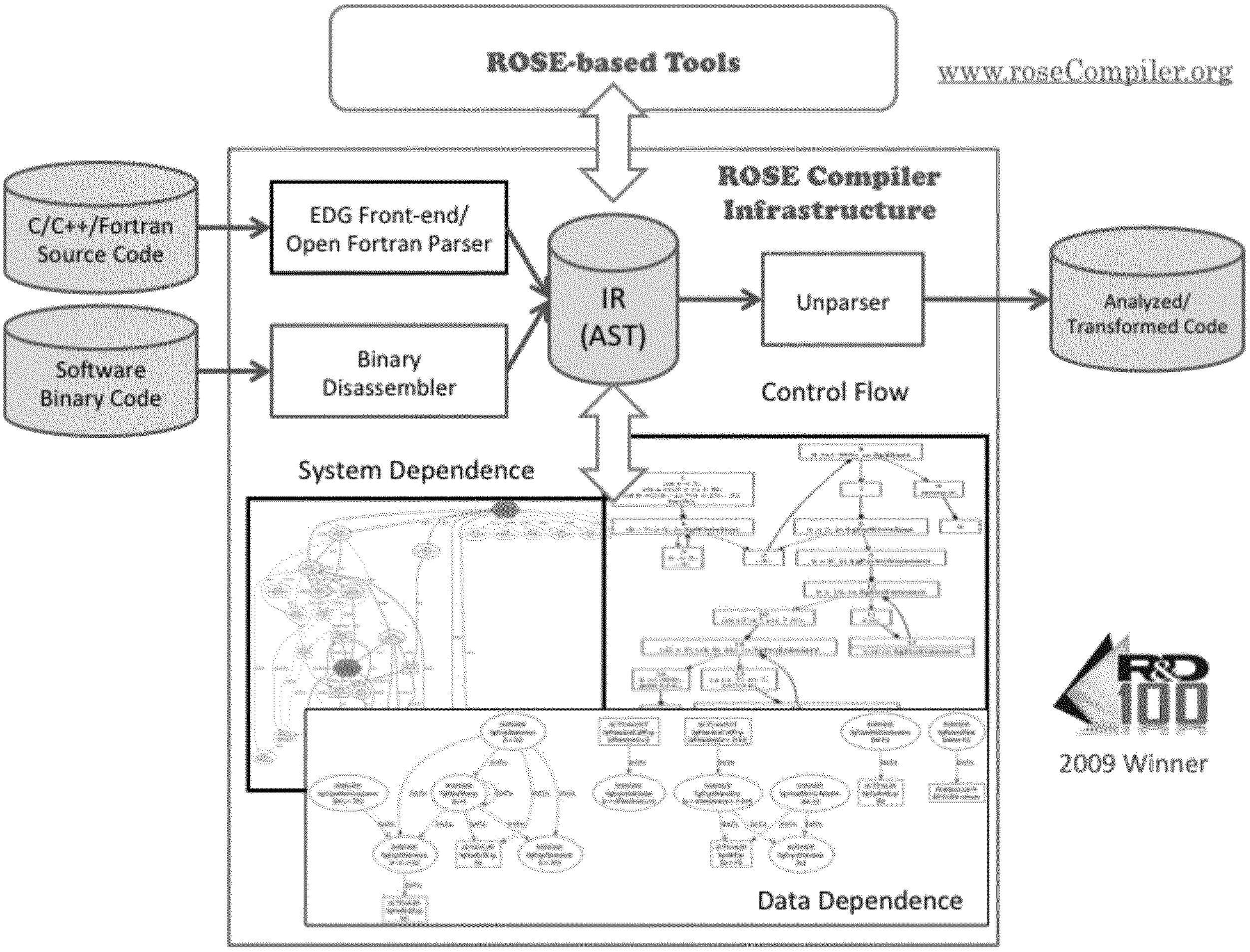


Figure 1: ROSE-Based Tools Constructed Using Infrastructure for Source Code and Binary Analysis

ROSE and Static Analysis

Compass is an extensible static analysis tool built on top of ROSE. It uses both source and binary analysis to find violations of structure-oriented rules. These rules can be either user-defined or taken from existing software assurance-related standards. Checkers for rules in Compass are implemented as plugins that interface to the core ROSE infrastructure. Thus, Compass provides an abstraction layer over complex analysis algorithms to simplify the process of writing custom plugins by tool developers. Currently, Compass includes approximately 150 checkers to find style, security, correctness, and portability vulnerabilities in both source code and binary executables. More Compass checkers have been built by the Software Engineering Institute (SEI), Carnegie Mellon University's Compu ter Emergency Response Team (CERT), NIST, and other research groups. Extending Compass to include more checkers for handling more sophisticated rules and using the analysis outlined in this proposal would provide an open source asset to the software assurance and electric power community and a basis for continued research work in the future.

2.4.1.2 The Application of Formal Methods

Formal methods is the name for a broad set of mathematically-based techniques for the specification, development, and verification of software and hardware systems. The over-arching goal of formal methods is to reason about program behavior for whole classes of scenarios. Alternative approaches to formal methods would be probing this behavior in a time-consuming, point-wise fashion and inaccurately inferring (often based on fairly broad assumptions) that these point tests are indicative of behaviors over whole regions of the state space. We plan to research and develop several formal methods techniques to support building ROSE-based software assurance tools: Satisfiability Modulo Theories (SMT) solver-driven model checking, symbolic execution, and abstract interpretation. Each of these general formal techniques can form the building blocks for specific classes of assurance-related analysis that will be developed as part of this proposal.

Boolean Satisfiability and SAT Solvers

A *Boolean formula* is a formula over the domain of Boolean true/false values. For example, in standard logic notation the formula ! $\vee (y \wedge \neg z)$ has as Boolean variables *x*, *y*, and *z*. The formula is true either if *x* is true, or else if *y* is true and *z* is false. Here the symbol ∨ represents logical disjunction (logical-or), ∧ represents logical conjunction (logical-and), and ¬ represents logical complementation (logical-not). Given a Boolean formula, a *Satisfiability solver* (SAT solver) is an algorithm for deciding whether there is a *satisfying assignment* of true or false to each Boolean variable such that the overall formula evaluates to true. For example, a satisfying assignment to the formula above would be $\{x \leftarrow false, y \leftarrow$!"#$, $z \leftarrow$! "#$%}.

Satisfiability Modulo Theories

The Satisfiability Modulo Theories (SMT) problem[16] is the determination of whether a logic formula in first-order logic is satisfiable. SMT solvers permit logic equations to be solved for satisfiability. Modern SMT solvers use at their core Boolean Satisfiability solver (SAT solver) algorithms to efficiently search and perform case analysis over large propositional formulas.

SMT Solvers

It is often better to translate a complex problem into a formula in a richer logic that can more naturally express the concepts of the problem domain, and then solve this formula directly. A *Satisfiability Modulo Theories solver* (SMT solver) is a SAT solver whose input language has been extended to encompass variables, terms, and predicates from a fixed collection of first order logic (FOL) theories. The collection of theories typically includes at least linear arithmetic over either the rationals or integers, symbolic bitvectors, arrays, and a theory of un-interpreted functions (a dynamically-generated theory of free function symbols drawn from the input formula). Introductions to the mathematical principles underlying SMT can be found at

For example, here is a simple SMT formula over the theories of linear integer arithmetic and un-interpreted functions: $f($!$) > y - 1 \wedge x <$! $\wedge f($!$(y))$

Any SMT solver that supports these theories would automatically conclude that the formula is unsatisfiable [12,14]. Although the formula is small and its validity is a simple consequence of the theories' axioms, already its proof requires quite a number of formal deduction steps in first order logic The number of needed proof steps increases dramatically for the much larger formulas that occur in real software verification, making automated SMT solvers essential.

SMT-based model checking is used to detect logic errors in software along each of the many paths of control flow in a program. An SMT-based model checker first translates a program into a logical formula that represents the program's input/output behavior. Any input that causes the formula to evaluate to the value *true* represents a program bug. An SMT solver is a tool that performs a search within the formula, looking for exactly such inputs. If it finds them, the inputs are then used to isolate which partof the program contains the defect.

Model Checking

Model checking refers to automatically testing whether a model of a system meets a given specification [17]. For software assurance, it permits the exploration of all paths in a program to enforce properties of specific abstractions defined as explicit state models. It takes a representation of the program as a state machine and seeks to enumerate the set of states that the program can possibly enter into. In particular, model checking is useful for identifying if any execution path can lead to specific classes of flaws like deadlock and related concurrency control bugs. Symbolic model checking explores a state space by manipulating logical expressions representing many states. Building state representations that carry rich semantic information in an automated (or, mostly automated) manner requires abstractions that are present in the source code [9,11]. Source code contains the higher level structure of a program in an unambiguous form.

Models derived from binary executables will be limited in the amount of high level semantic information that they carry with respect to the original program. The validity of the models is no different from those derived from the source, only the insight that can be gained from them with respect to the program logic itself. Thus when the source code is available, ROSE; which can represent the source structure without lowering will provide the highest level of support for such abstractions.

A different form of analysis, called *k*-bounded model checking, uses SAT and SMT solvers to perform checks generated from an analysis of each specific path (similar to the model checking described), but is able to focus on paths that are explicitly specified as most relevant and thus trade conservativeness in the analysis for performance. Our work will use this technique to prove narrowly defined properties of both source code and binary software (e.g., paths that would bypass security authentication; analysis for backdoors).

Symbolic Execution

Symbolic execution is an approach to simulating a program's execution by tracking and evaluating symbolic, abstract values rather than running the program with actual values in variables [28]. The purpose of symbolic execution is to reason about program behavior over entire ranges of values instead of point-sampling as common test-driven methods provide. It allows the software to be analyzed to understand its behavior under entire slices of the parameter space. Symbolic execution can explore execution paths and identify anomalies of values as it proceeds. This is because instead of executing the program as a compiled binary, the program is interpreted in a symbolic manner in which program primitives operate on abstract values, such as ranges or sets. This approach to analysis is a specialization of the more generic abstract interpretation method described above, and will support making assertions about program behavior over larger input spaces than a simple test suite is capable of.

Abstract Interpretation

A common source of program vulnerabilities is a failure to check that the inputs to subroutines or language primitives are well-formed. For example, a subroutine may have a numeric input that must fall within the range of 1-10 to execute correctly. Abstract interpretation is a general technique for automatically detecting violations of these kinds of input well-formedness requirements, by “abstractly” simulating the set of possible values each program variable can be assigned [18].

Abstract interpretation starts with the view of a program as a state transformer—at any given point in time, the program consumes inputs, produces outputs, and potentially modifies some internal state. A symbolic representation of the program is used to advance a representation of the state space such as ranges of values associated with a variable. This allows whole cohorts of states to be reasoned about simultaneously versus an approach that enumerates the individual execution trajectories explicitly. Each operation in this symbolic form of the program operates on these state representations and an interpreter is used to then observe how they change over time.

2.4.1.3 Building Analysis Tools

Formal methods can be directly used to build novel software assurance tools. They can also be used to improve classic compiler analysis so any existing tools built using such compiler analysis can indirectly benefit from formal methods. Our focus will be on the construction of unique tools tailored to the requirements of the electric power industry and supply chain integrity. We will be able to leverage a wide range of technologies available in ROSE to support these new forms of software analysis on both source code and binaries.

Static analysis will be most easily delivered using the existing Compass tool infrastructure built using ROSE. Compass will be extended to include new *checkers* that will leverage the new analysis capabilities. For example, symbolic analysis will permit values used as inputs to standard functions to be computed and range checked against their definitions. This approach will permit the detection of range checking problems via a static analysis. Checkers will be built to demonstrate this and target domain-specific functions (e.g., applications or library specific, where the user has defined proper ranges for inputs via program annotations; or standard library functions where the range of well-defined inputs can be easily known). Our other forms of formal methods analysis will similarly have domains of error checking which define new opportunities for Compass checkers.

Distribution of Tools

It is envisioned that the code analysis tools developed in this project will be released as open source and available to the power industry vendor community as well as the laboratories performing testing and certification. For example, EPRI could incorporate the tools into their laboratory testing framework, or a utility could use them to analyze new software releases in their own internal test environment. Furthermore, fostering the formation of a technical community within the Common Criteria Evaluation and Validation (CCEV) program developed with MITRE and sponsored by NIST and the NSA, which is focused on cyber security for electric power industry products, is another potential application of the tools developed in this project. The code analysis capabilities could be used to verify security function requirements that are specified as part of the protection profile that electric power products would incorporate in their design. Any tools could be released or restricted to specific customers at the discretion of DOE in order to best support the electric power community. The ROSE team at LLNL has a long history of work with NIST, SEI, and MITRE on open tools using ROSE.

2.5 Hardware Reverse Engineering to Detect Supply Chain Injections

Integrated circuits are both literally and figuratively black boxes. They perform complex operations, with no visibility to the outside world, usually faster than any human can follow. This immense capability powers our world, but the lack of visibility also conceals a great weakness–we cannot tell what the IC is actually doing. For critical systems, this is a major concern. Our solution to the problem of fully understanding our IC performs an intelligent brute force exploration using the available I/O to determine all possible states to a pre-specified depth, identifies loops, terminators, and equivalent states, and then reduces the state set to a minimal set. This minimal set is a very close equivalent to the actual state machine the designers used to create the IC. From this, we can build up a picture of the attribution of the state machine, and determine if it is possible to infer what the IC was designed to do.

2.5.1 Background/Previous Work

Previous work has been performed in hardware assurance, but due to limitations in the testing environments and computing power, it has been infeasible to brute force recreate the state machine behind the IC until now. Hardware testing traditionally involves three different viewpoints: quality control, destructive reverse engineering, and non-destructive side channel explorations. The first of these, quality control, has been a part of the IC manufacturers' production line for a long time, and is a well-developed area. However, quality control only looks to see that the IC is functioning properly, without considering that extra and unwanted functionality may have been covertly added.

Destructive reverse engineering involves taking a chip apart, layer by tiny layer, taking pictures, and then reconstructing the IC from the transistor layout. This is the gold standard of hardware reverse engineering, and will continue to be so, because there is nothing you can hide. However, there are drawbacks to this method, starting with the high cost of both the equipment and personnel time required to perform the breakdown and then the re-buildup. Destructive reverse engineering is also just that – destructive. Once a chip has been taken apart, you cannot put it back together again and use it. In situations where the researcher has only one of the ICs, or cannot have assurance that all the chips in a batch are the same, destructive hardware reverse engineering is of much reduced use.

Non-destructive methods of investigating an IC exist which usually operate by taking measurements of side channel output from the IC. In this manner, researchers have been able to derive encryption keys and other critical data, as well as the location of major components (memory, ALU, etc.) on larger ICs. However, this does not help in the complete analysis of the logic. A side channel exploration can tell you when something is happening differently from the expected baseline, but not exactly what, and it also assumes a valid baseline to compare to.

Finally, previous work in a non-destructive method of characterization of the finite state machines (FSMs) that our ICs are built from has been proposed by Brutscheck, et. al. [1], which shows great potential but has severe drawbacks in the speed of the algorithm. This algorithm creates a tree based on all outputs that create a change in state and output to determine the FSM. In software-based variations, a 5-state Mealy-type FSM IC analysis could take up to a week. A more recent field-programmable gate array (FPGA)-based variation by Uting, et. Al . [2], reduced times to less than 30 seconds. This is still a prohibitive amount of time when scaled to larger ICs, as many real-life state machines will have many more states and many more possible inputs.

2.5.2 Technical Approach

1) **IC Pin Profiling –** The first step in our solution is to find, through basic electrostatic discharge (ESD) phenomena, which pins are connected, which direction the pins are (input or output), and where the reset pin is. We also need to determine the speed of the IC.
2) **State Machine Exploration** – Having found the input and outputs, we then begin pushing those inputs, giving the IC a wide range of input strings, and recording the outputs that are produced. This creates a state machine tree, which is our initial map of the IC.
3) **State Machine Tree Reduction –** Having generated all possible states to a certain depth, we then apply various techniques to reduce the tree and match identical states. When the tree has all leaf nodes looping back upwards, or terminating, and all possible states have been matched, we have the final state machine.
4) **Repeat State Machine Exploration and Tree Reduction –** The state machine exploration is done in pieces, allowing us to process for terminations and reducing the problem space. However,

this approach then needs to loop, exploring then reducing, repeatedly, until the terminating conditions of step 3 are completed.

5) **State Machine Attribution** – After the terminating conditions are met and the final state machine produced, we can thenbegin to attribute parts of the state machine to various real-world outputs.

Example

An example might be the exploration of a simple IC with 1 input and 1 output pin. This method takes and tries all possible input combinations 7 deep (an arbitrary value) and records the output (e.g., input-output 0-1, 1-1, 1-1, 0-0, 0-0, 1-1, 0-1). These values are used to create a tree structure. In this example, the tree will have a binary structure, but in an IC with more pins or more possible inputs, this would correspondingly be a quad, oct, or larger tree. We can then find terminators (states where the output of any following state is the same), and 1 sided loops (states that ignore some inputs, remaining in that state until a specific input is given). For the purposes of this example, we have limited ourselves to loops that are only one state repeated; larger loops of multiple states repeated are feasible with some scaling. After finding the loops and terminators, assumptions can be made that some states are duplicates, based on their outputs and sub-nodes. After reducing the tree through these techniques, the original state machine is discovered.

The proposed solution is advanced through the use of intelligent brute forcing. This is accomplished by assigning an input stream length (7 long, in our previous example) that is long enough to allow for certain levels of confidence in the detection of terminators and smaller loops, but no further, to reduce computation time. An analysis is done on this initial tree, and the tree pruned to reducethe number of leaf nodes. After reduction, the limited set of leaf nodes are then expanded into their own full secondary trees for exploration and reduction, as well as inclusion with the overall tree for exploration of larger loops. This dramatically reduces the computation time of the exploration and analysis.

2.5.3 Physical System

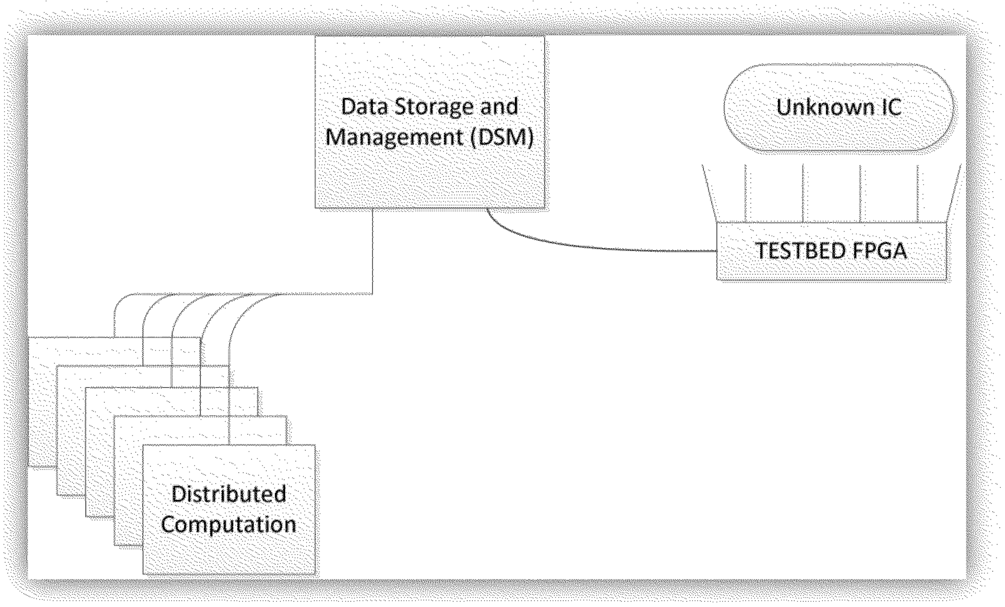


Figure 2:Logical diagram for hardware reverse engineering

The physical system is composed of three parts:

- Data Storage and Management (DSM) – stores original tree, modified versions, passes off chunks for evaluation to the distributed tree reduction parts; sends instructions for new lines of exploration to the hardware query part
- Testbed – part that actually talks to the test IC ; this will likely be the bottleneck, because we are limited to the speed of the IC itself
- Distributed computation –actually several machines; initially commodity servers will be used, but eventually customized hardware, pattern processors, FPGAs, or custom application-specific integrated circuits (ASICs) could be integrated to speed up the pattern detection.

These three parts need to be managed well, and possibly a flexible management system will need to be set up to keep the testbed and test IC part going at maximum efficiency.

2.5.4 Challenges

Speed

For this solution to be applicable, it must be able to do large ICs quickly, on the order of hours, rather than the days to weeks previous work has accomplished. We have several techniques that will be employed to gain the extra speed necessary for feasibility.

- Dividing the load – There are two bottlenecks in this system ; the first is on the processing end. When the tree has been explored, it must be reduced. This reduction will be shifted over to a distributed computing system, where the tree can be subdivided, explored and reduced, recombined, and then explored and reduced more. This subdivision of labor allows us to process the trees as fast as the testbed can bring in the data from the IC.
- Custom testbed – The second bottleneck is at the testbed itself. Most smaller ICs that are clock - based run at speeds closer to 50MHz, as opposed to the gigahertz speeds that we are used to in our processors. This causes the investigation to be slowed down to the speed of the IC. We will mitigate as much of this as possible by creating a custom testbed environment that can detect the speed of the IC and act accordingly. Also, by creating the testbed IC interface from hardware, we are able to move at speeds similar to the IC, without waiting for an interruptible software process.
- Hardware or other custom pattern detection solutions – As the tree is passed to the distributed system for computing, what is really being done is the detection of patterns in the data strings. There are several options, both in custom pattern detection languages like Snobol, and hardware such as the Pattern Processor, by Rick Dove [3], that will be implemented to increase the speed of the data processing.

We need confidence levels for two major reasons. The first is the depth of exploration; we are assuming that there may be infinite loops in the state machine, leading to a tree that is likewise infinite. Infinite loop => infinite tree => infinite time. To make time non-infinite, we need to do some arbitrary cutoffs in the tree. In the previous example, we used a cutoff of three ; a loop had to be repeated three times, or for a terminator it needed all paths downwards from it to have the same output for at least three further inputs, or for matching states, all sub-states of the two had to match for three deep. Because we must have some way of cutting off the state tree, we must likewise have some way of recording and evaluating the depth to which we cut things off. This is the confidence level.

We also need confidence levels for the possible passive applications of the technology. In the passive situation, we cannot, for whatever reason, actually interfere with the workings of the IC (possibly it is already installed in a system, etc.) and are forced to passively watch the normal operations of the IC. In normal operations, the IC will not likely try all possible inputs, but will have repeated patterns and input sets to goes through. We would like to develop an algorithm flexible enough to take in these inputs, flesh out the state tree as far as possible, and then try to work back to a state machine. Because the state tree will have many holes in it, we will need a confidence level to tell us just how full that tree actually was, and then how accurate the state machine we build from the state tree is.

Attribution

Attribution is a ubiquitous and challenging problem, and as such, we will be approaching it from two different directions. First, we will expand the work done by previous researchers [5] of destructive hardware reverse engineering in building up logic form the transistor layer, but in our case, we will begin at the state machine layer. Second, we will be creating small state machines that replicate or model given components (adders, I/O controllers, etc.), and creating a database of the state machine implementations. From this database, we can then take unknown state machines and compare them to determine similarity.

Computational Limitations

While this solution provides a critical function in the detection of Trojan or other malicious hardware [4], it does not solve all problems with our IC supply chain. It does not address non-state machine based ICs, as we assume that most combinational logic circuits can be explored without using this level of investigation. It also does not address ICs with residual memory. For the analysis of these ICs to work, we must have a reset state, a known state to return base our explorations from, and residual memory may change the reset state and any behavior afterwards. Finally, what we cannot detect with this method are very large loops in state machines. As we described in the Confidence Levels section, detection of the difference between a loop that cycles through 100,000 times and one which is infinite is very difficult and time-intensive to detect. While it may become feasible at a later state, with faster ICs and more intensive computing, very large loops are out of the scope of this project.

Possible Variations

There are several possible variations that could be followed after the initial 3-years research and development. The first of these is a portable version of the system, which, using customized hardware, could be capable of detecting and analyzing smaller state machines on-location wherever the IC may be in use. It may also be possible to profile ICs, rather than completely explore them, based on the timing between the state transitions. It would also be possible, through the same methods, to determine if there were extra states inserted in the state machine without output as an attempt to hide functionality. Finally, as we discussed in Confidence Levels, this system could be used in a passive way, as in the portable version, to analyze ICs that are currently running in systems.

2.6 Multi-lab Collaborative Technical Approaches

The three collaborating laboratories have complementary expertise to bring to the different parts of supply chain analysis. However, we have identified parts of the supply chain analysis that are outside of the specialized expertise of any one lab. Specific aspects that we identify below are uniquely suited to the combined and collaborative research expertise of the labs included in our proposal. These areas technically span software, trusted computing design goals, and hardware reverse engineering, and require the combined expertise across labs to develop new technologies, mathematics, or analysis breakthroughs required for supply chain analysis and future concepts of integrity.

2.6.1 ORNL-LLNL

ORNL and LLNL will work together to understand, define, analyze, test, and demonstrate a trust model (TM) that can be used by the electric power industry for the secure development and delivery of products in their supply chain. The verification of the correct implementation of the trust anchor (TA) is critical to building in *trust* to cyber-devices. Formal methods to prove specific properties of the implementation in the firmware are an essential technology to support this work and will be developed by ORNL-LLNL collaborations using ROSE. Additionally, the analysis of software using the TA is critical to ensure the implementation of trusted computing within applications. ORNL and LLNL will collaborate to build these tools for source code and binaries so that applications' use of the TA and thus the TM is supported correctly. Neither lab could expect to have the required expertise to support this work alone; we expect this is a substantially unique capability of this proposal, and a direct result of the team selected for this research work.

2.6.2 PNNL-LLNL

PNNL and LLNL will work together on how to support the reverse engineering of ICs. This is expected to be a significant goal requiring some computational expertise specific to parallel algorithms and formal method-based proof techniques. Solutions in this area may use some formal methods and parallel algorithm expertise that we expect will be required to improve the performance of the analysis algorithms to be developed. Existing expertise using SMT solvers for software analysis is expected to be of value in the computational aspects of the currently defined IC reverse engineering (RE) research. This collaboration is another unique opportunity that is a result of the selection of our team in this proposal. Interestingly, the origin of the formal methods research work used in ROSE at LLNL for software verification is derived from a few decades of research from hardware verification research. We will work together to integrate our understandings of the hardware reverse engineering research to more efficiently pose and solve some of the outstanding computational problems using SAT and SMT solvers. This joint work will be explored with LLNL assisting PNNL with technical expertise wherever possible. We expect this collaboration to result in unique research directions in how it will combine both SAT/SMT and HPC parallel algorithm research.

2.6.3 PNNL-ORNL

PNNL and ORNL will work together to define and articulate the policy space and context for supply chain security, and specifically integrity. Furthermore, PNNL will assist ORNL in articulating the context scope and impact of supply chain integrity efforts undertaken in this project. PNNL will provide insight and guidance when needed on the direction and crafting of Task 3.2: Supply Chain Architecture and Policy Analysis.

ORNL will advise PNNL on the role of trusted computing in architecture and policy. Furthermore, ORNL will direct the application of trusted computing in hardware for the PNNL lead tasks, specifically Task 3.7: IC Pin Profiling and State Machine Investigation, State machine tree attribution. ORNL's involvement in the Trusted Computing Group will provide insight and leadership in cutting edge directions for secure computing and device integrity.

2.7 Management of SCI-FI: Supply Chain Integration for Integrity

PNNL will lead the project while providing research and development on the hardware related tasks, testing and demonstrating solutions in the powerNET testbed, and overall project management. LLNL will lead the research and development efforts specific to software for both source code and binary executables, as well as development, testing, and release of Compass and other tools using new forms of analysis that will be tailored to supply chain analysis and the subject of this proposal. ORNL will lead the research on trusted computing architecture and policy research and development. Figure 4 shows the associated schedule for the tasks detailed above.

LLNL will work with ORNL to support the use of ROSE in tools they will develop; and work with PNNL on the formal methods technologies to investigate hardware IC evaluation. Utilizing an iterative development method, we will work together to integrate these new analysis techniques with existing Compass capabilities or other separate tools. We will demonstrate them on supply chain test cases that are identified in conjunction with technical personnel and specific electric power industry expertise on our team (at LLNL, PNNL, and ORNL). An automated and iterative release process will define comprehensive testing and make our tools available to others within our proposal team and whatever subset of the open release deemed appropriate by DOE. All work will be able to run on desktop computers; more advanced forms of proposed work may optionally include super computer resources. The project team has many years of experience releasing large-scale software open source software for use by universities, DOE labs, other US government agencies, and industry groups worldwide.

Our work will result in an integrated solution with static analysis and formal methods for source code and binary executables, firmware, and TC devices. By providing an integrated software assurance solution for the entire supply chain "stack," the proposed work will directly meet the requirements of our research work for supply chain integrity. Other parts of the proposal use the software capabilities to define integrated supply chain research with both hardware and trusted computing architectures.

We will also continue to release our tools under an open source, BSD-style license so they can be used for both open source and commercial users, as applicable. The resulting tools will be tested and demonstrated by analyzing large code bases to be selected as part of initial work in the proposal.

3 Tasks

3.1 ORNL-LLNL collaboration for Built-in Supply Chain Integrity

3.1.1 Task Description

ORNL and LLNL will work together to understand, define, analyze, test, and demonstrate a trust model (TM) that can be used by the electric power industry for the secure development and delivery of products in their supply chain. ORNL will research the current state of supply chain security and define a TM that can be applied to the power industry and respective vendor community to ensure the integrity and authenticity of delivered products, such as cyber-physical devices and intelligent electronic devices (CPDs/IEDs) used for substation automation systems (SAS), and wide area monitoring protection & control (WAMP&C).

The high-level implementation of the trust model will be verified and handled for both source code and binaries. This work will deliver three different tools to support the verification of the trust model. This analysis work is fundamental to the developed trust model being adopted as a basis for future supply chain security management.

Low-level support for the TCB selected will have to be evaluated using specialized tools that can perform verification on the binary firmware to ensure consistency and compliance with the specification. This lower-level verification pertains to how the cryptographic processing is performed and how the critical trust functions related to key management are handled.

If we assume that each device has some non-migratable and discoverable root of trust, then we have a way to build a trusted computing environment, beginning with device manufacture. ORNL and DMI will research a way to use trusted computing metrics to discover and identify violations of supply chain integrity. ORNL-DMI will define the TM specifications, and develop a verification and validation tool for firmware, source code, and device subcomponents utilizing system and subsystem TCBs. ORNL will then develop a suite of analysis tools to verify conformance with the TM and assurance of the cryptographic processing, with the assistance of the partner teams . Both ORNL and LLNL will work with Pacific Gas and Electric (PG&E) to determine how the selected TM can be applied to their product development and delivery process in order to provide enhanced security. Discussions with PG&E will also guide how prototype implementations of the TM can be developed, tested, and demonstrated.

3.1.2 Problem Addressed

Many CPD manufacturers do not provide effective protection against vulnerabilities in devices created by misbehaving components introduced in the supply chain. However, subcomponents of these devices typically have a trusted computing base (TCB) that contains a root of trust (RoT) or trust anchor (TA) and immutable measurement capabilities. For those CPDs that have a TCB, we can utilize the TCB to discover and track supply chain integrity and to prevent integrity failures. For example, changes to IED/CPD hardware or firmware can occur anywhere in the post-manufacture chain, including at the customer site. An adversary could:

- Apply changes to device software
- Apply a patch to device firmware
- Add or remove components to the device

Under current circumstances, it is likely that none of these actions would be detected by the customer. Once a compromise has occurred, it is possible that no security measure applied by the customer in deployment would be effective in mitigating the security exposure created by the compromise.

3.1.3 Background

Security of the power grid must begin with the manufacture of the components used to build and maintain the power grid. An economical trust must be developed for the purpose of creating integrity in the manufacture and delivery of equipment deployed into the power gridThe development of the TM is part of what must be verified as a foundation for firmware, hardware, and applications that run on top of the TCB. This work is the lower level verification of the software that comprises the TA.

3.1.4 Scope and Technical Approach

We will use ROSE as infrastructure for building the three specific tools for this task. More details on ROSE and its support for both source code and binary analysis can be found in section 2.4.

3.1.5 Milestones

Develop a trust model that can be applied to the electric power industry supply chain. Develop tools to analyze and verify the trust model and integrity of trust anchor (TA) aware hardware components, e.g., SoCs, CPUs, subcomponents, software layers, and both source code and binary executables. Demonstrate these tools with partner PG&E and determine how they can be incorporated into their technology roadmap.

Milestones for this work will be associated with the development and demonstration of these separate tools on selected firmware (both binaries and source code).

3.1.6 Deliverables

1) Requirements specification for the trust model – This specification will outline a common trust model architecture and provide a clear description to vendors for the capabilities and interfaces required to implement the trust model. This will also serve as the basis for testing and verification that a particular implementation meets the trust model. Included in the specification will be requirements for the trust anchor, the measurement capabilities on the platform, key management throughout the lifecycle of the keys, and system components that will be measured and verified during the verification process.
2) Tool for the verification and validation of disparate trusted anchor implementations. This tool will prove that the actual implementation is trustworthy (extract from binaries).
3) Leverage the TA architecture to create a verification and validation tool for
   a) Versioning control
   b) Provisioning via trusted computing services provisioning markup language
   c) Digital rights management
   d) Non-compiled and compiled software for correct and trusted functionality
   e) Hardware

3.2 Supply Chain Architecture and Policy Analysis

3.2.1 Task Description

Leveraging ORNL's experience with TPM/TCB/ RoT to provide context for hardware and software supply chain efforts, we will develop a trust architecture that can be applied to the electric power industry supply chain. We will also develop policies by analyzing TM and TA leveraging TRoT. Furthermore, we will develop a trust schema that will be used in a X.500 architecture with X.509 certificates to create an implementable architecture for supply chain integrity for the electric delivery system. This will aidPG&E and others to create value-added products that can be used to track and discover supply chain fraud, abuse, and malicious intent. We will also work with PG&E to determine how they can be incorporated into their technology roadmap.

3.2.2 Problem Addressed

An important realization in the energy delivery system is that the manufacturing and distribution infrastructure of today's computing devices are inadequate to assist in the establishment of trustand integrity of purchased products. As such, the cyber-physical industries that manufacture and distribute devices for our electrical infrastructures need to be augmented in such a way that it allows the conveying of relevant integrity information from one end of the supply chain (manufacturing) to the other (consumer).

3.2.3 Background

We propose the Integrity Management Model (IMM) for EDS. The IMM brings together important concepts and constructs. These include the notion of trust and provenance oftrusted platform components, the notion of trust scores, and how they play a role in a verifier's evaluation of a platform, the relevant XML schemas, and the general use of credentials backed by supporting integrity information.

3.2.4 Scope and Technical Approach

Technical assumption: We will have a trusted computing base. ARM, PowerPC, Intel, AMD, along with Microsoft, and Red Hat all have built in TAs. Albeit these anchors are not necessarily standardized, we are still able to leverage the TCB. We will use the notion ofsupply chain (networked) TRoT. Figure 3 represents the TRoT for a single cyber-physical device. Extending this to multiple devices and using Trusted Network Connect (TNC) we are able create a trusted architecture that will ultimately help to create a policy for a TCB-aware system.

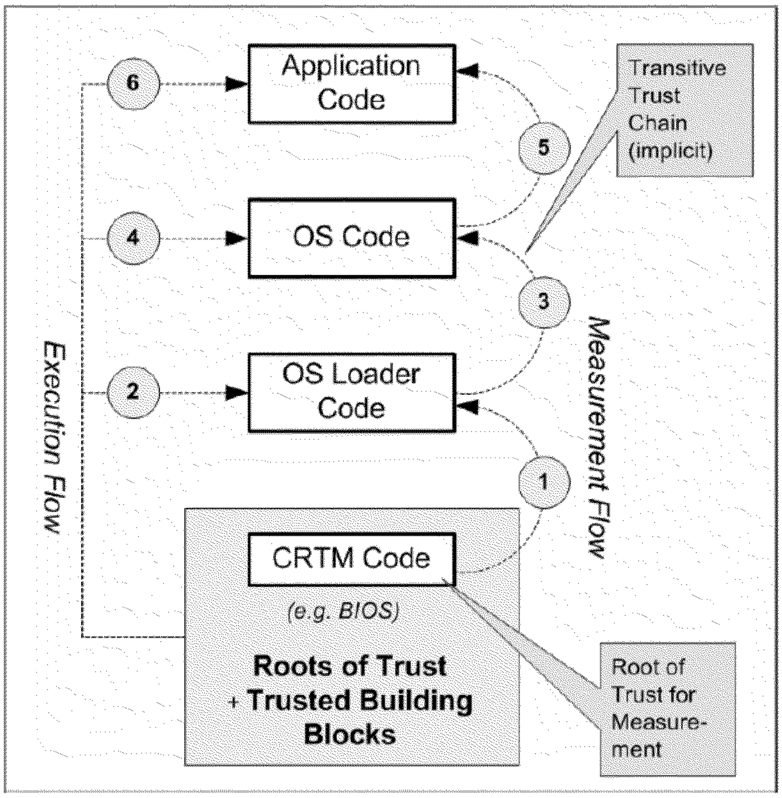

6
Application Code
Transitive Trust Chain (implicit)
5
4
OS Code
3
2
OS Loader Code
1
Execution Flow
Measurement Flow
CRTM Code
(e.g. BIOS)
Roots of Trust + Trusted Building Blocks
Root of Trust for Measure-ment

Figure 3: Infrastructure Working Group TCG: Trust Chain Emanating from RoT

The goal of integrity management is to record the measurements performed by components participating in a trust chain (i.e., sequence of component measurements) in a manner that is coherent for reporting to the external world (e.g., a verifying entity). Related to this is the correct handling of the trust of chain information within an IED and when it is reported to the external world. Protecting the recorded trust chain information of a CPD (e.g., against unauthorized modification) is also crucial since the trust chain is the basis for according trust (by an external entity) to that platform. A TA must be able to report its transitive trust chain in an unhindered manner lest it loses its core value of being a trustworthy platform.

3.2.5 Milestones

The ORNL team will create architecture of trust for the electric delivery system by creating TRoT applications that will do the following for the energy user.

1) Create integrity schemas. The integrity schema for EDS represent a set of documents that pertain to the representation of integrity-related information (syntax and semantics) in the XML schema format. Both static component integrity information (e.g., reference measurements) and dynamic integrity information generated by the platform at runtime (i.e., runtime measurements) are captured and represented by the schema specifications.
2) Create platform trust services (PTS) specification.
3) Create PTS communications via TNC (Interface-Map (IF-M)).
4) Create integrity information submission and publication interfaces.

3.2.6 Deliverables

1) Create policies for EDS TRoT
2) Create specifications for EDS TRoT
3) Policy and specifications for EDS TRoT
4) Go/No Go Decision {Will policies and specifications map to an implementable value add tool?}
5) Prototype TRoT for cyber-physical device
6) Prototype trusted network for TRoT for "n" cyber-physical devices (homogeneous)
7) Demonstrate a prototype trusted networked TRoT with actual CPDs
8) Go/No Go Decision {Is the TRoT system scalable to multi nodal system?}
9) Go/No Go Decision {Is the speed of TRoT below 10 seconds on non-energy devices?}
10) Work with PG&E to understand the specifications for a value-added tool for PG&E and other utilities

3.3 Extend ROSE for Analysis of Energy Infrastructure

3.3.1 Task Description

Some basic work is required to enable the development of the capabilities that will be leveraged for the work in this proposal (tasks 3.5 and 3.6). We expect that all work wil leverage existing work in ROSE (developed over the last dozen years and funded by DOE ASCR). Our proposed work will also leverage an existing tool built on top of ROSE (Compass), and include work extending the static analysis support in ROSE; additional work will add missing pieces required to support the work specific to the requirements of embedded devices for the electric power industry (e.g., how to load the executables for analysis, connection to external (assumed) hardware features, etc.). We will leverage the PG&E partnership and the multi-vendor PNNL powerNET testbed environment to provide guidance on target processors for power automation field devices.

3.3.2 Problem Addressed

This task addressed how to tailor the existing analysis in ROSE to the specifics of embedded devices for the electric power industry.

3.3.3 Background Scope and Technical Approach

Our team has an extensive background in the development of custom software analysis for both x86 and PowerPC Instruction set architectures. We have also supported the ARM instruction set in ROSE (required for embedded devices). ROSE support for ARM is not at the same level as that for x86 and PowerPC, so we will do more testing and development specific to the ARM instruction set.

3.3.4 Milestones & Deliverables

1) Develop a test suite of example binaries using the ARM instruction set (we expect to increase the number of test codes that we presently use for regression testing in ROSE).
2) Add instruction set semantics and execution control flow support for ARM instructions to ROSE. This is a low-level but critical feature important to the representation of logic required for SAT

and SMT solvers within proof techniques used to evaluate ARM binaries and detect vulnerabilities.

3) Add support for conversion of binary instructions to the low-level virtual machine (LLVM) compiler framework as a way to simplify new forms of analysis that we will develop and also leverage some existing forms of analysis. This work will also permit us to evaluate third party LLVM-based tools for binary analysis (previously used only for source code).
4) Add firmware analysis support to emulate the loader required to map the binary to memory and start execution. This step is a required part of the binary analysis to get the context of the binary in memory as part of static analysis that uses emulation and proof checking with mixed concrete and symbolic data derived from the load process.
5) ROSE can use the instruction set semantics to emulate and even symbolically evaluate binaries as part of a static analysis. This capability exists in ROSE for x86 and PowerPC; this work will be extended to ARM as part of the development of the ARM instruction set semantics to support this sophisticated level of analysis for embedded device code (binaries). This will permit a number of the forms of binary analysis to be applied to ARM-based binaries to build some of the tool deliverables listed previously.
6) Analyze binaries in ROSE symbolically and with mixed concrete and symbolic support starting at arbitrary offsets into the code. This approach will be extended to ARM binaries as required to support the evaluation of patches as part of the delivery of tools for assessing supply chain integrity.

3.4 Develop Tools for Analyzing Firmware Releases of Electric Power Industry Field Devices

3.4.1 Task Description

Develop tools for analyzing firmware releases of electric power industry field devices.

3.4.2 Problem Addressed

EPRI has identified as a significant concern of their member organizations the authenticity, integrity, and assurance of field device firmware that is supplied by their vendors. In order to address some of these concerns, a set of tools that can perform code analysis of the vendor firmware releases will be developed.

3.4.3 Background, Scope and Technical Approach

Our approach will be to work with our partner, PG&E, to develop tools to evaluate field devices in an automation lab setting and obtain firmware releases for that equipment from the vendors. Example field devices are meters, protective relays, digital fault reorders, RTUs, etc. The tools developed will be provided to PG&E for their use in security testing and analysis. We will develop tools to verify using proof technologies (as represented by SAT and SMT equation generation and solvers) and a wide range of properties of the firmware; we will work with PG&E and others to define which types of properties are most critically important.We will also develop more checkers in Compass for detection of both source code and binary vulnerabilities (using the formal methods technologies developed in ROSE). We will extend the existing formal methods technologies in ROSE to support more tailored forms of analysis for the electric power industry, supply chain evaluation, SCADA systems, and embedded software generally. Specific work will focus on the evaluation of patches; this work is especially difficult because of the arbitrary starting points analysis requires, but technologies in ROSE have addressed this concern in previous binary analysis work making ROSE especially well-suited to this form of patch analysis.

3.4.4 Milestones & Deliverables

1) Select a particular field device for which a firmware release/update is available for analysis We will select specific devices upon which to focus initial work based on guidance from partner PG&E and the vendor equipment available in the PNNL powerNET testbed in order to define initial tools. We will then iterate on the release of these tools using an expanded selection of devices. This will ensure that our work is always relevant and demonstrable.
2) Identify topics for most critical software analysis. Part of the initial work will be to identify the most critical software issues for supply chain analysis and how it relates to hardware to form a complete picture of supply chain vulnerabilities.
3) Develop a tool for detecting the presence of anti-disassembly technologies. These technologies are extremely well correlated to both advanced forms of malware and IP protection–neither of which should be expected in firmware updates to power utility equipment. This work will ensure that all software developed for use within power utilities is fundamentally analyzable. Any software detected as having these technologies could still be analyzed using ROSE, but its presence is especially suspicious. Such techniques would invalidate the analysis using other more common binary analysis tools such as IDA Pro.
4) Develop a tool for analyzing differences between two firmware releases and prove techniques to evaluate the differences. This work will use state-of-the-art algorithms recently developed in ROSE for the evaluation of the differences between two large binaries. The technology will scale to handle realistic binaries and be coupled with analysis to evaluate each difference between any two binaries (such as that represented by a firmware update).
5) Develop a tool for detecting the presence of unused code. The existence of unused code can be a mechanism to obscure the behavior and complicate the analysis of both source code and binaries. We will develop thetechniques required to support an evaluation of all software for unused code. Initial work to support this is already present and will be extended in ROSE and developed into a separate tool.

3.5 Develop Tools for Analyzing SCADA Application Software

3.5.1 Task Description

We will develop tools for analyzing SCADA software used to monitor and perform supervisory control of power grid operations. Software monitoring and reacting to inputs and output within the SCADA system are at the level of application software (higher level than that of firmware embedded on the field devices). Verifying the software at this level is useful at both the source code and binary executable level since either may be available (and the source code and binaries each have specific advantages and disadvantages within the software verification process). This task addresses this critical level in our integrated approach to supporting supply chain analysis and measuring its integrity.

3.5.2 Problem Addressed

Mission criticalEMS applications such as SCADA software run on general-purpose operating systems and commercial server platforms and are used by utilities to monitor electric power generation, transmission, and distribution. In order to address the supply chain concerns associated with development, delivery and implementation of critical SCADA application software, we will develop a set of tools that can perform automated code analysis.

3.5.3 Background, Scope, and Technical Approach

Our approach will be to initially work with open source SCADA software to define, develop, and test techniques for performing automated code analysis. We will then pursue working with the vendor software available in the PNNL powerNET testbed to apply these tools to a larger set of SCADA application binaries.

3.5.4 Milestones & Deliverables

1) Select an open source SCADA product for internal tool development. Initial work will start will the selection of concrete examples of open SCADA work so that we can specialize the software analysis in ROSE to target problems specific to SCADA. We will also evaluate SCADA examples using the existing forms of analysis and Compass checkers.
2) Develop a tool for analyzing unintended execution paths. To support more aggressive testing of source code and binaries, we will define path feasibility as a way of restricting the number of paths in the software over which to conduct deeper forms of analysis. This makes the evaluation process more efficient and also focuses any deeper analysis using super computer resources to be more effective and eliminates numerous forms of false positives from being reported.
3) Develop a tool for mapping inputs and outputs. Because of the design of SCADA software to process many inputs and outputs, we will focus on forms of analysis to map how inputs are treated in the software (including identification of paths that can reset inputs and outputs and bypass how input and output values are communicated to operators).
4) Develop a tool to detect the presence of backdoors; code paths that can bypass known security authentication can be a security problem. Such paths are famous for having been detected in software as part of vendor support, but are also significant security problems. Without analysis of the software it is impossible to verify the absence of such backdoors.

3.6 LLNL-PNNL Collaboration to Apply Formal Methods Approaches to Hardware Analysis

3.6.1 Task Description

The reverse engineering of IC is expected to be a significant goal requiring some computational expertise specific to parallel algorithms and formal method-based proof techniques. Both LLNL and PNNL will work together; LLNL will support this work with some formal methods and parallel algorithm expertise that is expected to improve the performance of the analysis algorithms to be developed. Existing expertise using SMT solvers for software analysis is expected to be of value in the computational aspects of the currently defined IC RE research.

3.6.2 Problem Addressed

The formal methods applied to reverse engineering hardware will come in two parts. The first is that, when verifying hardware, much like when verifying software, we need to be able to prove the basic rules for verification hold true. If these basic building blocks cannot be proven, the higher level assumptions cannot be trusted. Secondly, we believe that a formal methods analysis, especially in conjunction with the parallel algorithm experience of LLNL, will allow us to develop better code to divide, reduce, and recombine the state machine tree described in Section3.8.

3.6.3 Background, Scope and Technical Approach

The origin of the formal methods research work used in ROSE at LLNL for software verification is from hardware verification research. The techniques used to evaluate software are deeply connected to verification technology that synthesizes hardware from software and uses these techniques, originally reserved for hardware verification, to prove properties of software. We expect that there are modest connections between techniques understood at LLNL and technologies being developed at PNNL. This joint work will be explored with LLNL assisting with technical expertise wherever possible.

Formal methods will be used in two areas in the hardware analysis: verification of reduction and attribution methods and developing efficiency in the distributed computing algorithms. Both of these tasks are large, complicated, and will be intertwined with and supporting other, more concrete tasks. The formals methods analysis of our attribution and verification methods will be delivered as results for the milestones described below.

3.6.4 Milestones and Deliverables

1) Verification of basic reduction techniques
2) Analysis and implementation of formal methods analysis of distributed computing algorithms for speed and efficiency increases
3) Verification of advanced reduction techniques
4) Analysis of attribution methods

3.7 IC Pin Profiling and State Machine Investigation

3.7.1 Task Description

The first step in determining an IC's functionality is to determine which pins are connected (there are often multiple separate sections on one IC), and which are input and which are output. This has been done, and has been well documented, using basic ESD phenomenon. We also need to find a reset pin, which can be found by trying each pin as a starter to a basic mini-tree exploration, then retrying it. If the mini-trees are the same, we can assume that the pin is reset. Another method is power-off, power-on, but this is likely to be slower than a regular reset pin.

The second step in determining the state machine upon which an IC is built is filling out a state tree of all possible input-output streams. For a single-input IC, this tree is binary; for a two-input-pin IC, the tree would be a quad tree, and so on. This creates some very interesting explosion (exponential) of possible paths. From this tree, we can use reduction techniques and formal methods to reduce and prove identical states, bringing us to an approximation of the original state machine.

3.7.2 Problem Addressed

To understand a state machine-based IC, the state machine must first be explored This task accomplishes both the exploration of the physical hardware to determine the input and output pins, as well as the exploration of the state machine.

3.7.3 Background, Scope, and Technical Approach

Prior work by the PNNL team has successfully performed simple variations of this system, focusing on very small ICs. We will continue improving and enlarging this system to allow for more speed and larger systems, iteratively improving, evaluating, and researching.

3.7.4 Milestones and Deliverables

1) Physical system in place that allows for the attachment of a testbed and real ICs
2) Demonstration of successful pin profiling
3) Demonstration of pin profiling and state machine investigation on a small IC in less than a day of processing
4) Demonstration of working distributed processing system
5) Addition of confidence level into output
6) Demonstration of small IC in less than an hour and medium IC in less thana day
7) Demonstration of medium ICs in less than an hour.

3.8 State Machine Tree Attribution

3.8.1 Task Description

Once we have the state machine, the next step is to work upwards to what each part of the state machine is supposed to do (what input does it take in, how does it process it, does it control some output and what does that output actually do?). We will be building on work done by prior researchers, who build upwards from the transistor layer provided by the destructive hardware reverse engineering.

3.8.2 Problem Addressed

An IC needs to be reverse engineered and then its intent deduced. By attributing components of the state machine the IC is built upon, we can begin to understandits intent. This task will provide a more formalized method for attributing the state machine, starting with small components or pieces of the state machine and then using those to build larger components.

3.8.3 Background, Scope and Technical Approach

Prior work has been performed from the transistor level in this field [5], and we will be building upon this research. We will first attempt to identify smallercomponents (adders, check for condition loops, etc.) and using those smaller components to build up larger components (e.g., I/O controller). Secondly, we will create small state machines that replicate or model given components (adders, I/O controllers, etc.) and create a database of the state machine implementations. From this database, we can then take unknown state machines and compare them to determine similarity. These attribution methods will be formally verified, as per task 3.7.

3.8.4 Milestones and Deliverables

1) Demonstrate ability to identify smaller components or state machine pieces of 1 to 5 states
2) Demonstrate ability to identify state machine pieces of approximately between 5 and 20 states
3) Demonstrate ability to assemble larger components from smaller components
4) Demonstrate ability to begin associating real-world output from components. This milestone will require the application and testing of an IC with intention of placing it, after state machine exploration and identification, in a real-world system.

3.9 Develop, Execute, Test, and Evaluate Proposed Solution

3.9.1 Task Description

PNNL, with assistance from ORNL, LLNL , and PG&E will design, develop, and implement the test supply chain solutions, within the powerNET testbed, where applicable. When executed, the implemented tools will automatically configure the test suite environment to the appropriate start state for the test case. In addition, the powerNET collaboration portal will provide the information and documentation to execute the test case for the current tool as well as the procedure to execute the test for any team partner to independently test and evaluate.

3.9.2 Problem Addressed

To provide an efficient test framework and environment for SCI-FI, the testbed must provide a realistic scenario design and setup. To provide a useful testbed, it must automatically configure the equipment to the appropriate start state as well as provide the necessary material to execute the test.

3.9.3 Background

PNNL's powerNET testbed has been used in testing critical infrastructure components, synchrophasor research and development, control system communication, and electrical grid distribution simulation development. This team, utilizing the powerNET testbed, is capable of providing SCI-FI the required capabilities for the successful completion of the "Detect Compromise of Supply Chain Integrity" project area.

3.9.4 Milestones & Deliverables

1) Team develops and reviews environment design description
2) PNNL implements architecture for testing in powerNET testbed
3) PNNL and team begin testing and evaluation
4) PNNL and team complete testing and evaluation
5) Team drafts results report

4 Works Cited

[1] Cppcheck. http://cppcheck.sourceforge.net/.

[2] CVS - Common Vulnerabilities and Exposures. http://cve.mitre.org/.

[3] Dmalloc. http://dmalloc.com/.

[4] Dynist: An Application Program Interface (API) for Runtime Code Generation. http://www.dyninst.org/.

[5] FindBugs - Find bugs in Java programs. http://findbugs.sourceforge.net/.

[6] Open Fortran parser. http://fortran-parser.sourceforge.net/.

[7] Parasoft delivers quality as a continuous process. http://www.parasoft.com.

[8] Pmd. http://pmd.sourceforge.net/.

[9] PReach : A distributed model checker for Murphi. https://bitbucket.org/jderick/ preach/overview.

[10] SMT-LIB: The Satisfiability Modulo Theories Library. http://www.smtlib.org/.

[11] Spin - Formal Verification. http://spinroot.com/spin/whatispin.html.

[12] The Alt-Ergo theorem prover. http://alt-ergo.lri.fr/.

[13] Valgrind. http://valgrind.org/.

[14] Yices: An SMT Solver. http://yices.csl.sri.com/.

[15] CWE: Common Weakness Enumeration, A community-Developed Dictionary of Software Weakness Types. http://cwe.mitre.org/index.html, 2005.

[16] C. Barrett, R. Sebastiani, S.A. Seshia, and C. Tinelli. Satisfiability modulo theories. Handbook of Satisfiability, 185:825–885, 2009.

[17] E. Clarke. Model checking. In Foundations of Software Technology and Theoretical Computer Science, pages 54–56. Springer, 1997.

[18] P. Cousot and R. Cousot. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fix-points. In Conference Record of the Fourth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, pages 238–252, Los Angeles, California, 1977. ACM Press, New York, NY.

[19] Coverity Inc. Coverity static code analysis for c/c++, c#, and java. http://www.coverity.com/products/static-analysis.html.

[20] Edison Design Group. C++ front-end. http://www.edg.com.

[21] K.J. Hayman. An analysis of ordnance software using the MALPAS tools. Technical report, Director Electronic Research Laboratory, 1992.

[22] Hex-Rays SA. IDA Pro Disassembler and Debugger. http://www.hex-rays.com/ idapro/.

[23] Intel Corporation. Inspector xe 2011 from intel. http://software.intel.com/en-us/ articles/intel-inspector-xe/.

[24] Intel Corporation. Static security analysis (SSA) from Intel. http://software.intel.com/ en-us/articles/static-security-analysis/.

[25] International Business Machines Corp. IBM software - software analyzer. http://www-01. ibm.com/software/awdtools/swanalyzer/.

[26] Stephen Johnson. Lint, a C program checker. Technical report, Bell Laboratories, 1977.

[27] KindSoftware. Esc/java2. http://kind.ucd.ie/products/opensource/ESCJava2/.

[28] James C. King. Symbolic execution and program testing. Commun. ACM, 19:385–394, July 1976.

[29] Klockwork Inc. Klockwork insight. http://www.klocwork.com/products/insight.

[30] Quinlan. ROSE Compiler Infrastructure. http://www.rosecompiler.org/.

[31] Software Engineering Institute, Carnegie Mellon University. The CERT C Secure Coding Standard, Version 2.0. https://www.securecoding.cert.org/confluence/x/HQE, 2010.

[32] The MathWorks, Inc. Polyspace Embedded Software Verification for C/C++ and Ada. http: //www.mathworks.com/products/polyspace/index.html.

[33] Veracode, Inc. Application Security — Veracode. http://www.veracode.com/.

[34] Recent GAO report on threats to the IT supply chain: http://www.gao.gov/products/GAO-12-579T

[35] Article "Embedded system security much more dangerous, costly than traditional software vulnerabilities": http://www.csoonline.com/article/704346/embedded-system-security-much-more-dangerous-costly-than-traditional-software-vulnerabilities

[36] Occupying the Information High Ground: Chinese Capabilities for Computer Network Operations and Cyber Espionage: http://www.uscc.gov/RFP/2012/USCC%20Report_Chinese_CapabilitiesforComputer_NetworkOperationsandCyberEspionage.pdf

[37] Optimization and Implementation of a Nonlinear Identification Procedure of Unknown ICs. Brutscheck, Schmidt, Frank, Schwarzbacker, St. Becker. ISSC 2009

[38] Identification of Deterministic Sequential Finite State Machines in Unknown CMOS ICs Brutscheck, Schmidt, Frank, Schwarzbacker, St. Becker. ISSC 2010

[39] FPGA based Optimization and Implementation of Nondestructive Identification Procedures. Uting, Brutscheck, Becker, and Schwartbacher. ISSC 2011

[40] Pattern Recognition without Tradeoffs: Scalable Accuracy with No Impact on Speed. Dove. Cybersecurity Applications and Technology Conference for Homeland Security 2009

[41] Experiences in Hardware Trojan Design and Implementation. Jin, Kupp, Makris. HOST 2009

[42] Reverse Engineering: And Industrial Perspective. Vinesh Raja, Kiran Jude Fernandes, 2008 Google eBook

5 Project Management Plan

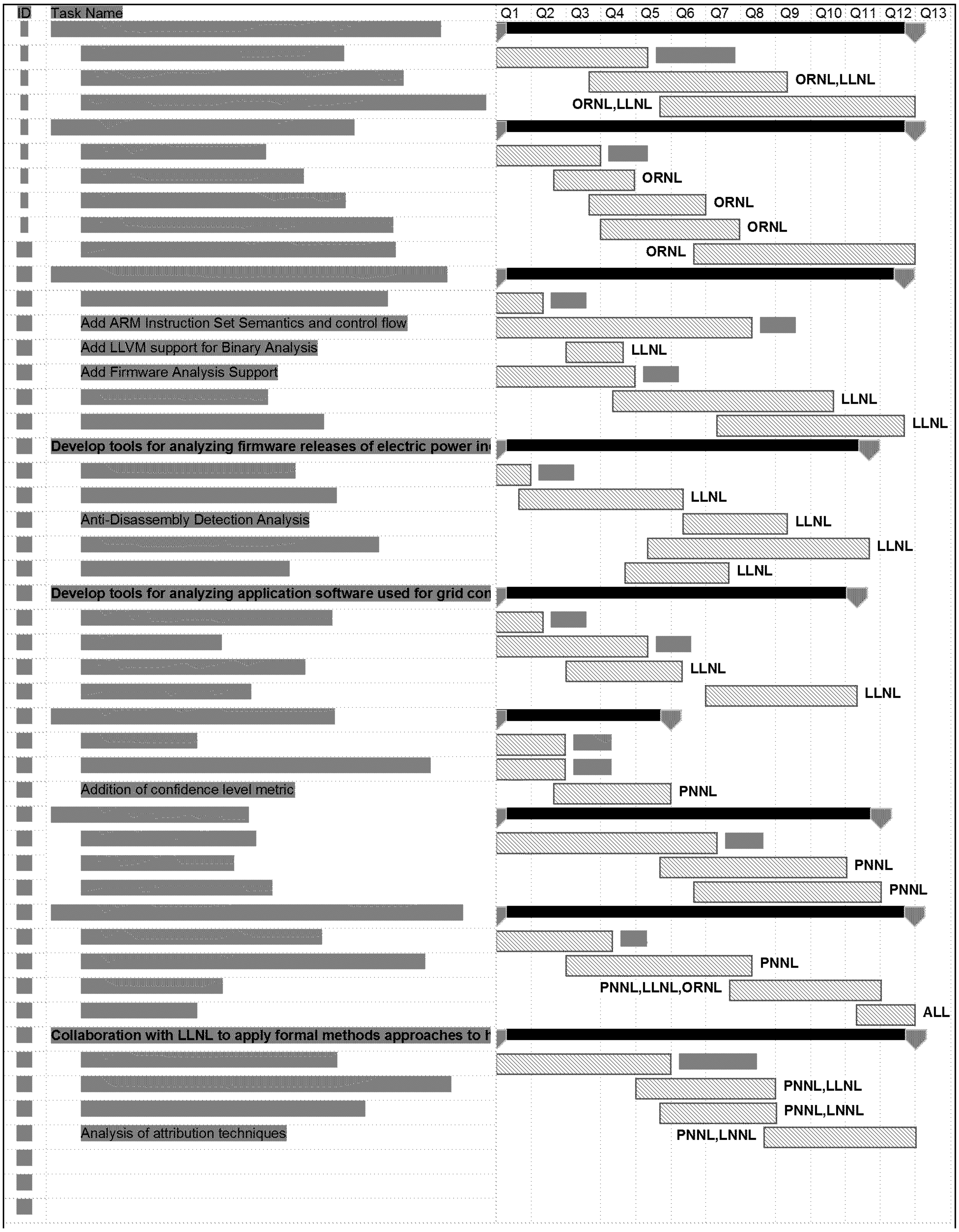


Figure 4: Research Tasks and Deliverables Schedule

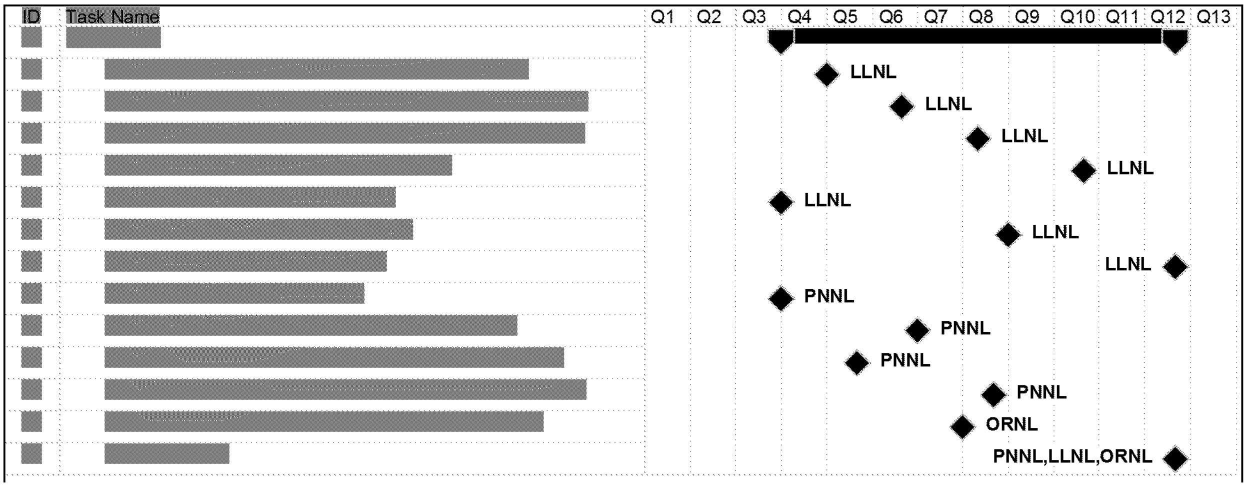
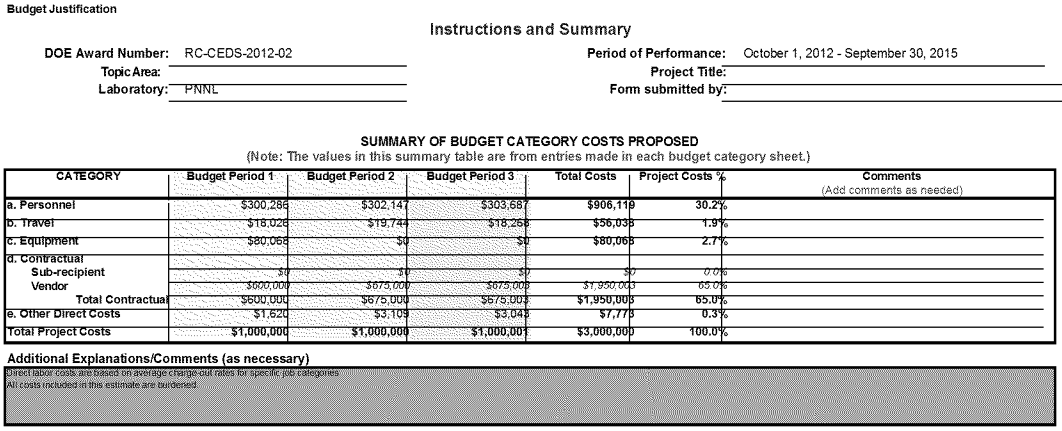


Figure 5: Research Tasks and Deliverables Schedule (cont.)

Our management plan contains tasks, specific deliverables, and milestones. We will additionally have teleconferences and face-to-face meetings to support our collaboration. Separate meetings with PG&E will allow us to tailor the work to best meet practical requirements.

6 Budget Justification



Budget Justification

Instructions and Summary

DOE Award Number: RC-CEDS-2012-02

Topic Area:

Laboratory: PNNL

Period of Performance: October 1, 2012 - September 30, 2015

Project Title:

Form submitted by:

SUMMARY OF BUDGET CATEGORY COSTS PROPOSED

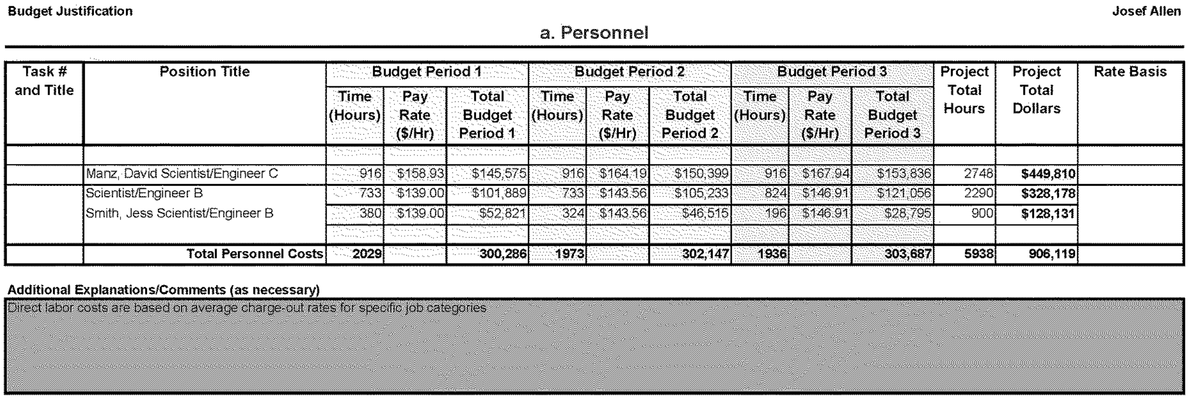
(Note: The values in this summary table are from entries made in each budget category sheet.)

CATEGORY	Budget Period 1	Budget Period 2	Budget Period 3	Total Costs	Project Costs %	Comments (Add comments as needed)
a. Personnel	$300,286	$302,147	$303,687	$906,119	30.2%	
b. Travel	$18,026	$19,744	$18,268	$56,038	1.9%	
c. Equipment	$80,068	$0	$0	$80,068	2.7%	
d. Contractual						
Sub-recipient	$0	$0	$0	$0	0.0%	
Vendor	$600,000	$675,000	$675,003	$1,950,003	65.0%	
Total Contractual	$600,000	$675,000	$675,003	$1,950,003	65.0%	
e. Other Direct Costs	$1,620	$3,109	$3,043	$7,773	0.3%	
Total Project Costs	$1,000,000	$1,000,000	$1,000,001	$3,000,000	100.0%	

Additional Explanations/Comments (as necessary)

Direct labor costs are based on average charge-out rates for specific job categories
All costs included in this estimate are burdened.

Figure 6: Budget Justification – Summary



Budget Justification

Josef Allen

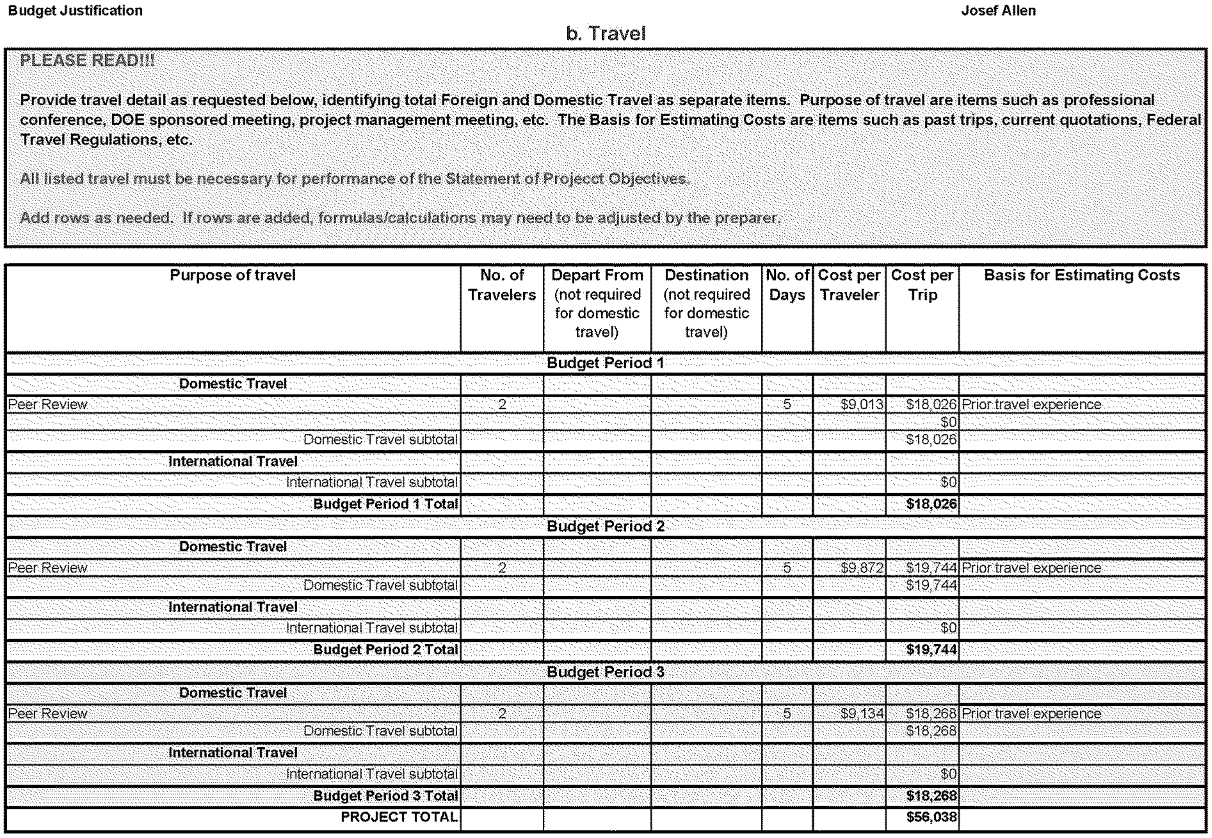
a. Personnel

Task # and Title	Position Title	Budget Period 1			Budget Period 2			Budget Period 3			Project Total Hours	Project Total Dollars	Rate Basis
		Time (Hours)	Pay Rate ($/Hr)	Total Budget Period 1	Time (Hours)	Pay Rate ($/Hr)	Total Budget Period 2	Time (Hours)	Pay Rate ($/Hr)	Total Budget Period 3			
	Manz, David Scientist/Engineer C	916	$158.93	$145,575	916	$164.19	$150,399	916	$167.94	$153,836	2748	**$449,810**	
	Scientist/Engineer B	733	$139.00	$101,889	733	$143.56	$105,233	824	$146.91	$121,056	2290	**$328,178**	
	Smith, Jess Scientist/Engineer B	380	$139.00	$52,821	324	$143.56	$46,515	196	$146.91	$28,795	900	**$128,131**	
	Total Personnel Costs	**2029**		**300,286**	**1973**		**302,147**	**1936**		**303,687**	**5938**	**906,119**	

Additional Explanations/Comments (as necessary)

Direct labor costs are based on average charge-out rates for specific job categories

Figure 7: Budget Justification – Personnel



Budget Justification

Josef Allen

b. Travel

PLEASE READ!!!

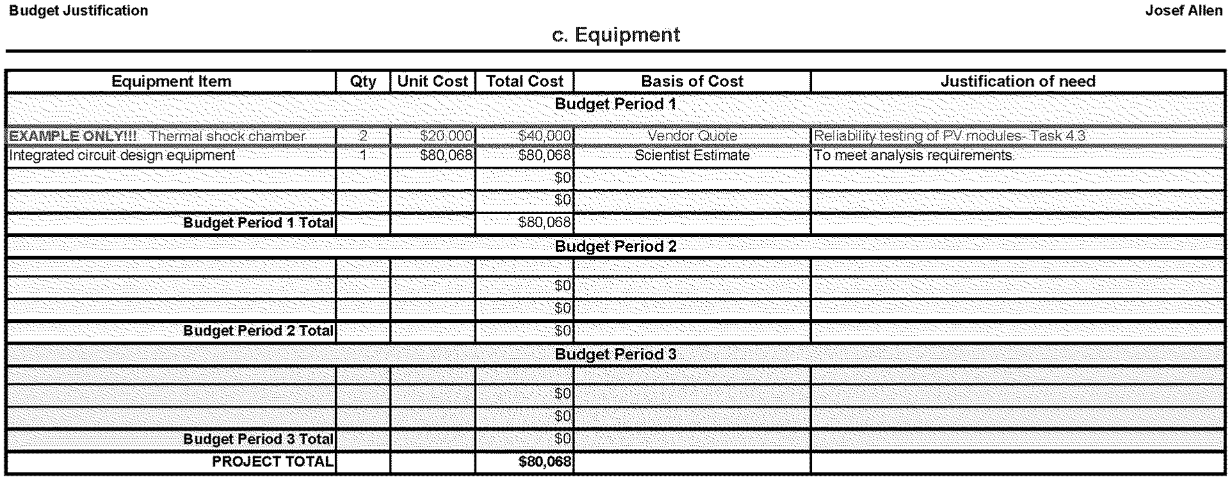
Provide travel detail as requested below, identifying total Foreign and Domestic Travel as separate items. Purpose of travel are items such as professional conference, DOE sponsored meeting, project management meeting, etc. The Basis for Estimating Costs are items such as past trips, current quotations, Federal Travel Regulations, etc.

All listed travel must be necessary for performance of the Statement of Projecct Objectives.

Add rows as needed. If rows are added, formulas/calculations may need to be adjusted by the preparer.

Purpose of travel	No. of Travelers	Depart From (not required for domestic travel)	Destination (not required for domestic travel)	No. of Days	Cost per Traveler	Cost per Trip	Basis for Estimating Costs
Budget Period 1							
Domestic Travel							
Peer Review	2			5	$9,013	$18,026	Prior travel experience
						$0	
Domestic Travel subtotal						$18,026	
International Travel							
International Travel subtotal						$0	
Budget Period 1 Total						**$18,026**	
Budget Period 2							
Domestic Travel							
Peer Review	2			5	$9,872	$19,744	Prior travel experience
Domestic Travel subtotal						$19,744	
International Travel							
International Travel subtotal						$0	
Budget Period 2 Total						**$19,744**	
Budget Period 3							
Domestic Travel							
Peer Review	2			5	$9,134	$18,268	Prior travel experience
Domestic Travel subtotal						$18,268	
International Travel							
International Travel subtotal						$0	
Budget Period 3 Total						**$18,268**	
PROJECT TOTAL						**$56,038**	

Figure 8: Budget Justification – Travel



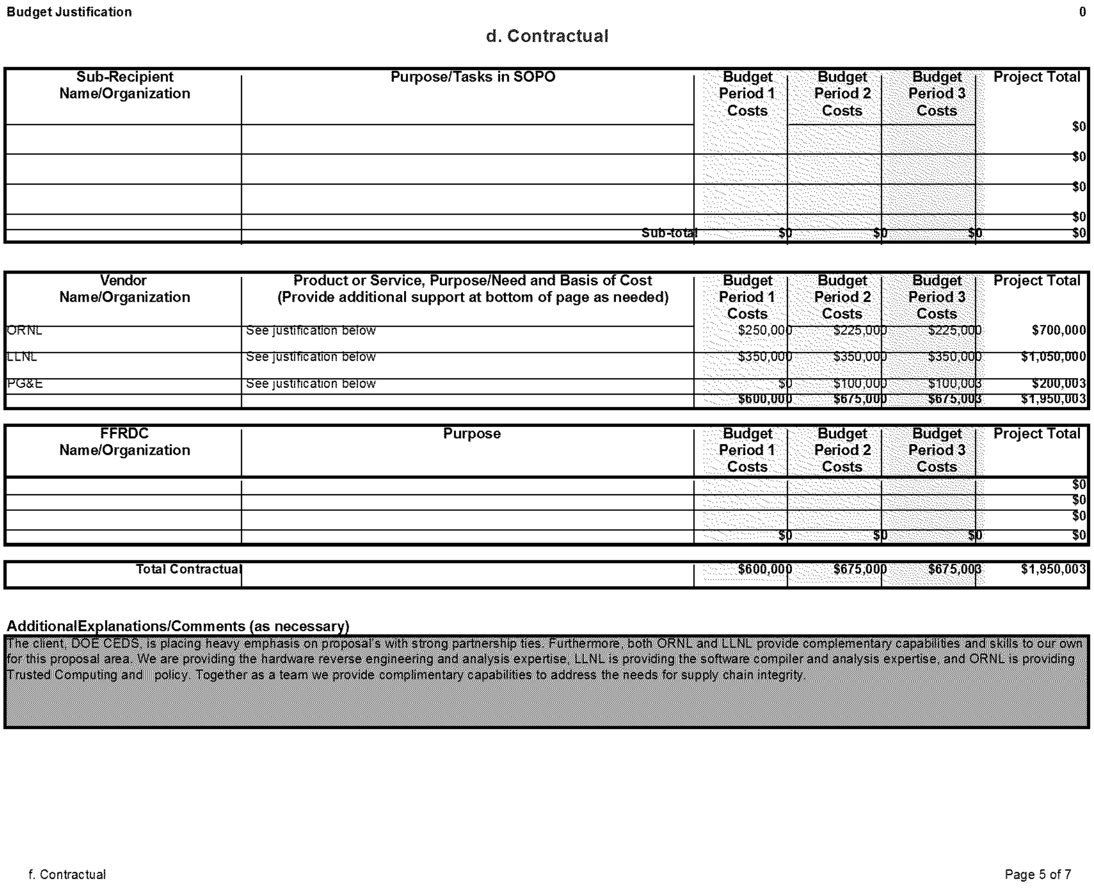
Budget Justification

Josef Allen

c. Equipment

Equipment Item	Qty	Unit Cost	Total Cost	Basis of Cost	Justification of need
Budget Period 1					
EXAMPLE ONLY!!! Thermal shock chamber	2	$20,000	$40,000	Vendor Quote	Reliability testing of PV modules- Task 4.3
Integrated circuit design equipment	1	$80,068	$80,068	Scientist Estimate	To meet analysis requirements.
			$0		
			$0		
Budget Period 1 Total			$80,068		
Budget Period 2					
			$0		
			$0		
Budget Period 2 Total			$0		
Budget Period 3					
			$0		
			$0		
Budget Period 3 Total			$0		
PROJECT TOTAL			**$80,068**		

Figure 9: Budget Justification – Equipment



Budget Justification 0

d. Contractual

Sub-Recipient Name/Organization	Purpose/Tasks in SOPO	Budget Period 1 Costs	Budget Period 2 Costs	Budget Period 3 Costs	Project Total
					$0
					$0
					$0
					$0
	Sub-total	$0	$0	$0	$0

Vendor Name/Organization	Product or Service, Purpose/Need and Basis of Cost (Provide additional support at bottom of page as needed)	Budget Period 1 Costs	Budget Period 2 Costs	Budget Period 3 Costs	Project Total
ORNL	See justification below	$250,000	$225,000	$225,000	$700,000
LLNL	See justification below	$350,000	$350,000	$350,000	$1,050,000
PG&E	See justification below	$0	$100,000	$100,003	$200,003
		$600,000	$675,000	$675,003	$1,950,003

FFRDC Name/Organization	Purpose	Budget Period 1 Costs	Budget Period 2 Costs	Budget Period 3 Costs	Project Total
					$0
					$0
					$0
		$0	$0	$0	$0

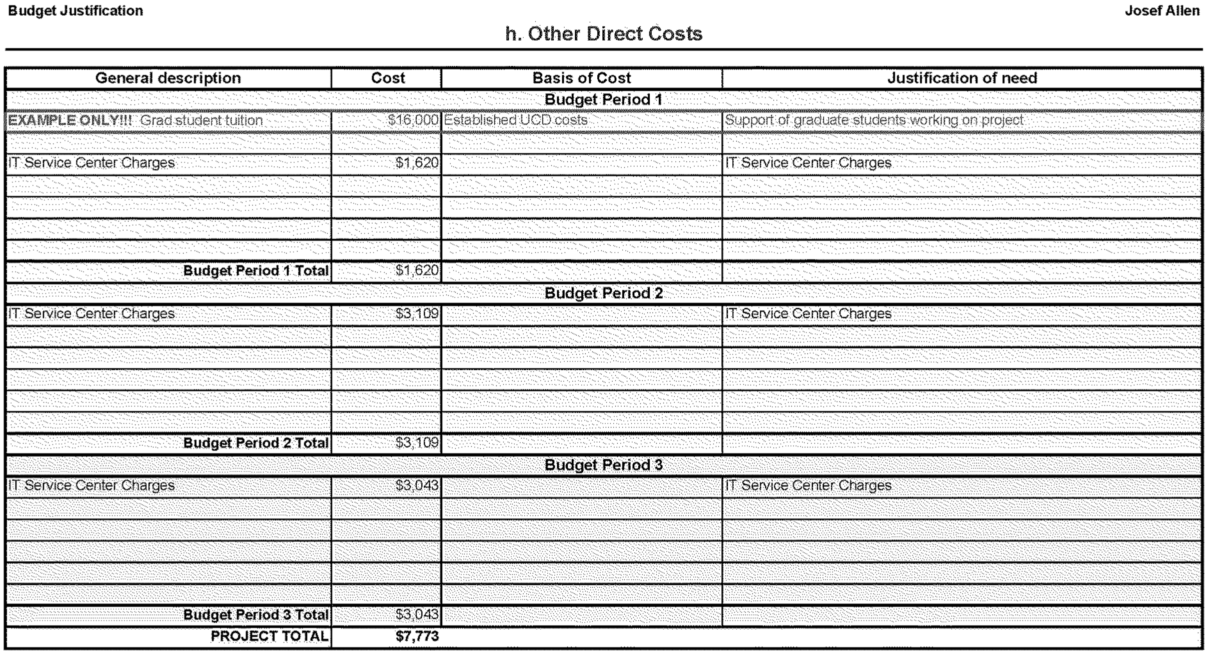
Total Contractual		$600,000	$675,000	$675,003	$1,950,003

AdditionalExplanations/Comments (as necessary)

The client, DOE CEDS, is placing heavy emphasis on proposal's with strong partnership ties. Furthermore, both ORNL and LLNL provide complementary capabilities and skills to our own for this proposal area. We are providing the hardware reverse engineering and analysis expertise, LLNL is providing the software compiler and analysis expertise, and ORNL is providing Trusted Computing and policy. Together as a team we provide complimentary capabilities to address the needs for supply chain integrity.

f. Contractual Page 5 of 7

Figure 10: Budget Justification – Contractual



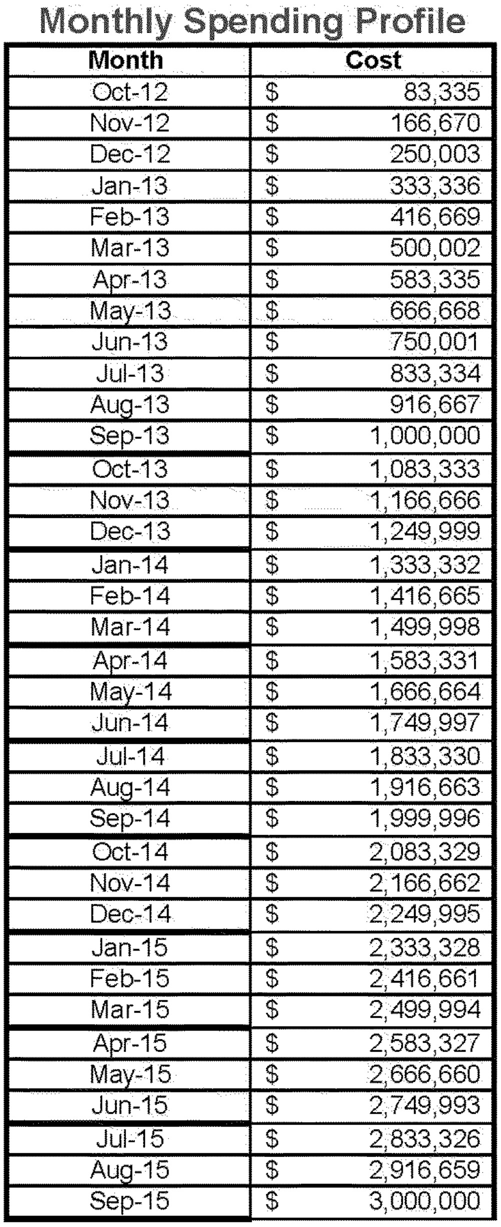
Budget Justification

Josef Allen

h. Other Direct Costs

General description	Cost	Basis of Cost	Justification of need
		Budget Period 1	
EXAMPLE ONLY!!! Grad student tuition	$16,000	Established UCD costs	Support of graduate students working on project
IT Service Center Charges	$1,620		IT Service Center Charges
Budget Period 1 Total	$1,620		
		Budget Period 2	
IT Service Center Charges	$3,109		IT Service Center Charges
Budget Period 2 Total	$3,109		
		Budget Period 3	
IT Service Center Charges	$3,043		IT Service Center Charges
Budget Period 3 Total	$3,043		
PROJECT TOTAL	**$7,773**		

Figure 11: Budget Justification – Other Direct Costs



Monthly Spending Profile

Month	Cost
Oct-12	$ 83,335
Nov-12	$ 166,670
Dec-12	$ 250,003
Jan-13	$ 333,336
Feb-13	$ 416,669
Mar-13	$ 500,002
Apr-13	$ 583,335
May-13	$ 666,668
Jun-13	$ 750,001
Jul-13	$ 833,334
Aug-13	$ 916,667
Sep-13	$ 1,000,000
Oct-13	$ 1,083,333
Nov-13	$ 1,166,666
Dec-13	$ 1,249,999
Jan-14	$ 1,333,332
Feb-14	$ 1,416,665
Mar-14	$ 1,499,998
Apr-14	$ 1,583,331
May-14	$ 1,666,664
Jun-14	$ 1,749,997
Jul-14	$ 1,833,330
Aug-14	$ 1,916,663
Sep-14	$ 1,999,996
Oct-14	$ 2,083,329
Nov-14	$ 2,166,662
Dec-14	$ 2,249,995
Jan-15	$ 2,333,328
Feb-15	$ 2,416,661
Mar-15	$ 2,499,994
Apr-15	$ 2,583,327
May-15	$ 2,666,660
Jun-15	$ 2,749,993
Jul-15	$ 2,833,326
Aug-15	$ 2,916,659
Sep-15	$ 3,000,000

Figure 12: Budget Justification – Monthly Spending Profile

7 Resume Files

David Manz, Project Director

Cyber Security Research Scientist, Secure Cyber Systems in the National Security Directorate, PNNL

Education and Training

University of Idaho	Computer Science	Ph.D., 2010
University of Idaho	Computer Science	M.S., 2005
University of Oregon	Computer Information Science	B.S., 2003

Summary of Qualifications

Dr. David Manz has been a Cyber Security Research Scientist at Pacific Northwest National Laboratory (PNNL) since January, 2010. His work at PNNL focuses on research projects for critical infrastructure, control systems, and information assurance. These projects include group key management protocols, secure control system communication, and emergency management. Dr. Manz has also contributed to the development and analysis of the Secure SCADA Communications Protocol and has assisted in human factors situational cyber security awareness research. Prior to his work at PNNL, Dr. Manz spent over five years as a Researcher on Group Key Management Protocols for the Center for Secure and Dependable Systems at the University of Idaho (UofI). In this position he researched Group Key Management Protocols on wireless and partially connected networks. He also received the NSF Cyber Corps Scholarship for Service Fellowship. In 2006, Dr. Manz held an internship with the Laboratory for Telecommunications Sciences at the University of Maryland where he worked in the field of Delay Tolerant Networks, explored topology generation and modeling, and evaluated open source tools to enable network protocol research, development, and testing. From 2001 to 2003, Dr. Manz was the Senior System Administrator for the Robert D. Clark Honors College at the University of Oregon where he was responsible for the entire network and student lab. Dr. Manz also has extensive experience teaching undergraduate and graduate Computer Science courses at UofI.

Professional Experience

- January 2010 – Present
  Research Scientist, Secure Cyber Systems in the National Security Directorate, PNNL
  Research projects for critical infrastructure, control systems, and information assurance. Contributed to the development and analysis of a Secure SCADA communication protocol, and assisted in human factors situational cyber security awareness research; also selected to participate in the prestigious Scientist and Engineer Development Program
- January 2008 – December 2009
  Senior Research Assistant, Department of Computer Science, UofI, Moscow, ID
  Culminated in a Ph.D. dissertation: Group Key Management Protocols on wireless and partially connected networks Center for Secure and Dependable Systems
- August 2005 – December 2007
  Scholarship for Service (SFS CyberCorps) Fellowship , Department of Computer Science, UofI , Moscow, ID
  Continued Ph.D. research in the field of cryptographic key management; extending Group Key Management into the wireless adhoc arena

Publications (selected)

DO Manz, TW Edgar, and TE Carroll. 2012. "Challenges of Cybersecurity Research in a Multi-user Cyber-Physical Testbed." Abstract submitted to NIST Cybersecurity for Cyber-Physical Systems Workshop. PNNL-SA-85616.

TW Edgar, **DO Manz**, and TE Carroll. 2012. "Towards an Experimental Testbed Facility for Cyber-Physical Security Research." In Proceedings of the 7th Annual Workshop on Cyber Security and Information Intelligence Research (CSIIRW 2011), October 12-14, 2011, Oak Ridge, Tennessee, ed. FT Sheldon, R Abercrombie and A Krings, p. Article No. 53. Association for Computing Machinery, New York, NY. doi:10.1145/2179298.2179357

Edgar TW, TE Carroll, **DO Manz**, and FL Greitzer. 2012. "Realizing Scientific Methods for Cyber Security." In CERT Learning from Authoritative Security Experiment Results (LASER) workshop. PNNL-SA-87207, Pacific Northwest National Laboratory, Richland, WA.

DO Manz and TW Edgar. 2010. “A Hybrid Authentication and Authorization Process for Control System Networks.” In *The International Conference on Information Assurance and Security.* PNNL-SA-72489, Pacific Northwest National Laboratory, Richland, WA.

TW Edgar, MD Hadley, **DO Manz**, and JD Winn. 2010. *Secure and Efficient Routable Control Systems.* PNNL-19474, Pacific Northwest National Laboratory, Richland, WA.

DO Manz, P Oman, and J Alves-Foss. “A Framework for Group Key Management Protocol Assessment Independent of View Synchrony.” *Journal of Computer Science* 6(3): 229-234, 2010.

DO Manz, J Alves-Foss, and S Zheng. 2007. “A Network Simulator for Group Key Management Algorithms.” *Journal Information Assurance and Security*, 2(2).

S Zheng, **DO Manz**, J Alves-Foss, and Y Chen. “Security and Performance of Group Key Agreement Protocols.” IASTED Networks and Communication Systems, March 2006, pp: 321-327.

K King, **DO Manz**, P Ortman, D Shikash io, and P Oman. “A Rapidly Reconfigurable Computer Lab for Software Engineering Security Experiments and Exercises.” Workshop on Secure Software Engineering Education and Training, in conjunction with the 19th Conference on Software Engineering Education and Training, (April 19-21, Turtle Bay, Oahu, HI), 2006.

S Caltagirone, P Ortman, S Melton, **DO Manz**, K King, and P Oman. “Design and Implementation of a Multi-use Attack-Defend Computer Security Lab.” Paper STISE09, Proceedings of the 39th Annual Hawaii International Conference on System Sciences, (January 4-7, Poipu, HI), 2006.

Synergistic Activities (selected)

- Key contributor to the DOE CEDS PNNL 61850 Acceleration effort
- Co-PI for the PNNL cyberNET testbed
- Key contributor to PNNL GridOPTICS powerNET testbed
- PM for the DHS Cyber-Physical Research Laboratory at PNNL

Josef Allen, Co-Principal Investigator

Education

Ph.D. in Applied Mathematics, Florida Institute of Technology, Melbourne, Florida 2007

M.S. in Computer Science/Security, the Florida State University, Tallahassee, Florida 2002

B.S. in Pure Mathematics, the Florida State University, Tallahassee, Florida 2000

Professional Training and Certifications

2002 – Present: Information Systems Security and Assurance for Professionals (INFOSEC) sponsored by the National Security Agency (NSA) and National Security Telecommunications Information Systems Security Committee (NSTISSC 4011)

2012 – Present: Invited Expert Program for Trusted Computing Group (TCG)

TCG is a not-for-profit organization formed to develop, define and promote open, vendor-neutral, industry standards for trusted computing building blocks and software interfaces across multiple platforms.

Relevant Work Experience

2012 to Present: Dr. Allen is a part of the Invited Experts Program for the Trusted Computing Group. TCG is a not-for-profit organization formed to develop, define and promote open, vendor-neutral, industry standards for trusted computing building blocks and software interfaces across multiple platforms.

2010 to Present: Senior Research & Development Staff Member, Oak Ridge National Laboratory (ORNL), Dr. Josef D. Allen is the Lead Scientist for Cyber -Physical Security for Critical Infrastructure Protection and Security for Oak Ridge National Laboratory. Dr. Allen's duties include Systems' Security Engineering, Secure TCP/IP for layers 3, 4 and 5 and Secure Network Architecture for SCADA and Substation Automation. He serves on the Department of Energy's Office of Electricity Cyber Security for the Electric Delivery Systems. He is also a member of the National SCADA Test Bed, which is comprised of several National Laboratories. Dr. Allen is the primary interface for ORNL as it pertains to cutting edge research cyber-physical systems and devices.

2002 to 2010: Previously he was Chief Systems Security Architect for Harris Corporation's Government Communications Systems Division. Duties included, Biometrics, Cyber Security, Digital Forensics Procuring, implementing and Architecting Distributed Systems for the IC and DOD Community. Enterprise System Security Architect Procuring, implementing and Architecting a Distributed systems. DCID and FIPS Pub 140 -2 compliant systems utilizing Linux , Windows, RTOS, Embedded Systems , Oracle, Enterprise Service Bus (ESB), Light Weight Directory Access Protocol (LDAP) X.500 protocol with Secure Labeling and Integrity utilizing Symmetric and Anti Symmetric Cryptography Protocols. Typical program fiscal responsibility was greater than or equal to $10 Million. All programs were fieldable into customer installations.

Honors and Awards

- 15 Awarded US Patents and 10 US Patents Pending
- NSA/NSTISSI INFOSEC Professional
- Combat Action Ribbon
- Kuwaiti Liberation Medal, By Kuwait
- Kuwaiti Liberation Medal, By Saudi Arabia

- South West Asia Service Medal with 3 stars
- Navy/USCG Unit Commendation Medal, Navy
- Meritorious Unit Commendation Medal
- Sea Service Deployment Ribbon
- National Defense Service Medal

Synergistic Activities

Professional Activities

2010 – Present: DOE/OE Cyber Security Electric Delivery Systems

2005 – Present: National Institute of Standards and Technology (NIST)

2010 – Present: DHS/ICSJWG Communications Subcommittee Member

Faculty Appointments

2010 – Present: Florida State University, Department of Computer Science
Areas of Study: Cyber-Physical Systems/Security Critical Infrastructure Protection, Steganography
Students: Sereyvathana Ty, M.S. Candidate in Digital Forensics (Graduated and Working at Sandia)
Joshua Lawrence, Ph.D. Candidate in Critical Infrastructure Protection
2010 – Present: Hampton University, School of Engineering and Technology
Areas of Study: Cyber Security, Critical Infrastructure Protection

Relevant Publications

Allen, J.D., Ty, S., Liu, X., Lozano I., *"Preventing Cascading Event: A Distributed Cyber-Physical Approach"*, CSIIRW '11 Proceedings of the Seventh Annual Workshop on Cyber Security and Information Intelligence Research Article No. 54 ACM New York, NY, USA ©2011 ISBN: 978-1-4503-0945-5

Michaels, A.J.,**Allen, J.D.**, *"Physical-Layer Encryption for Enhanced Cyber-Physical Security"*, CSIIRW '11 Proceedings of the Seventh Annual Workshop on Cyber Security and Information Intelligence Research Article No. 45 ACM New York, NY, USA ©2011 ISBN: 978-1-4503-0945-5

Allen, J.D., Liu, X., *"A Cyber-Physical Approach to a Wide-Area Actionable System for the Power Grid"*, MILCOM 2012; MILITARY COMMUNICATIONS CONFERENCE, 2012 (Pending Acceptance 02 Jul 2012)

Allen, J.D., Burmester, M., Easton, S., Guidry, D., Lawrence, J., Liu, X., Ty, S.' *"A Framework for Trusted IEC61850/62351 Cyber-Physical Devices for Substation Automation"*, MILCOM 2012; MILITARY COMMUNICATIONS CONFERENCE, 2012 (Pending Acceptance 02 Jul 2012)

Invited Conference Talks

Allen, J.D., Topics of Discussion were: *"Cyber-Physical Security Approaches for the Electric Delivery System"*, Invited to two DHS/ICSJWG Conferences, 2011 Spring & Fall

Liu, X. & **Allen, J.D.**, Topic of Discussion is: *"Cyber-Physical Security"*, MILCOM 2012 MILITARY COMMUNICATIONS CONFERENCE, 2012

Dan Quinlan, Co-Principal Investigator

Senior Research Computer Scientist at the Center for Applied Scientific Computing, LLNL.

Education and Training

• Los Alamos National Laboratory	Computational Mathematics	PostDoc, 1993
• University of Colorado at Denver	Applied Mathematics	Ph.D., 1993
• University of Colorado at Denver	Applied Mathematics summa cum laude	B.A., 1987

Summary of Qualifications

Dr. Daniel Quinlan has been a project leader for the ROSE project at LLNL. ROSE is an open source source-to-source compiler framework for building custom analysis and transformation tools for both source code (large scale applications in C, Fortran, C++, Java, OpenMP, UPC, Python, and other languages) and binary executables (x86, PowerPC, and ARM; for Linux and Windows). This work is focused on both the optimization of scientific codes, Software Assurance, and Cyber-Security specific forms of analysis. His research is in numerous areas that intersect Computer Science and Computational Mathematics. Research interests include: compilers, programming languages, Cyber-Security, computer architectures, parallel numerical algorithms, Adaptive Mesh Refinement, and Partial Differential Equations.

Professional Experience

1998 – Present
Computer Scientist at Lawrence Livermore National Laboratory

- Compile-time optimization and analysis of C, C++, Java, and F2003
- Program Analysis and Compiler Construction
- Cyber-Security
- Software Assurance

1994 – 1998
Acting Team Leader and Staff Scientist at Los Alamos National Laboratory

- Adaptive mesh refinement
- Numerical analysis

Publications (selected)

Hongyi Ma, Qichang Chen, and Liqiang Wang, Chunhua Liao and **Daniel Quinlan**, "OpenMP-Checker: Detecting Concurrency Errors of OpenMP Programs Using Hybrid Program Analysis", submitted to ICPP'12, The 41st International Conference on Parallel Processing, Pittsburgh, PA, September 10-13, 2012.

Jacob Lidman, **Daniel Quinlan**, Chunhua Liao, Sally A. McKee, "ROSE::FTTransform – A Source-to-Source Translation Framework for Exascale Fault-Tolerance Research", accepted by Fault-Tolerance for HPC at Extreme Scale (FTXS 2012), Boston, June 25-28, 2012.

Sara Royuela, Alejandro Duran, Chunhua Liao, **Daniel Quinlan**, "Auto-scoping for OpenMP tasks", accepted by the 8th International Workshop on OpenMP, IWOMP 2012, Rome, June 11-13, 2012.

Shah Mohammad Faizur Rahman, Jichi Guo, Akshatha Bhat, Carlos Garcia, Majedul Haque Sujon, Qingy Yi, Chunhua Liao, **Daniel Quinlan**, "Studying The Impact Of Application -level Optimizations On The Power Consumption Of Multi-Core Architectures", ACM International Conference on Computing Frontiers 2012 (CF'12), May 15th-17th, 2012, Cagliari, Italy.

Shalf, J. and **Quinlan, D.** and Janssen, C., "Rethinking Hardware-Software Codesign for Exascale Systems", Computer, Vol. 44, issue 11, pages 22-30; November 2011.

M.J. Sottile, C. Rasmussen, W.N. Weseloh, R.W. Robey, **D. Quinlan**, J. Overbey (2011). "ForOpenCL: Transformations Exploiting Array Syntax in Fortran for Accelerator Programming." Proceedings of the 2nd International Workshop on GPUs and Scientific Applications (GPUScA), Galveston Island, Texas. October, 2011.

Peter Pirkelbauer, Chunhua Liao, Thomas Panas, **Daniel Quinlan**, "Runtime Detection of C -style Errors in UPC code", 5th Conference on Partitioned Global Address Space Programming Models. October 2011.

Chunhua Liao, **Daniel Quinlan** and Thomas Panas, A ROSE-based OpenMP 3.0 Research Compiler Supporting Multiple Runtime Libraries, *International Workshop on OpenMP (IWOMP) 2010*, March. 2010

Chunhua Liao, **Daniel Quinlan**, Jeremiah J. Willcock and Thomas Panas, Semantic -Aware Automatic Parallelization of Modern Applications Using High-Level Abstractions, *Journal of Parallel Programming*, Jan. 2010

Guodong Li, Ganesh Gopalakrishnan, Robert M. Kirby, **Daniel Quinlan**; A Symbolic Verifier for CUDA Programs 15th ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming, January 9-14, 2010, Bangalore, India

Synergistic Activities (selected)

- The project leader for the ROSE project, an open source compile supporting custom analysis and optimization of large scale C, C++, Fortran, Java, UPC, CUDA, and OpenCL scientific applications within DOE
- Program committee member and reviewers for numerous conferences including: IBM Verification Conference'05, HIPS'07, HPCSW'08, PADTAD'08, PADTAD'09, IPDPS'10, CF2010, PADTAD'10, PACT11
- Invited talks: Google TechTalks

Ken Masica, Co-Principal Investigator

Electrical Engineer in the Systems Engineering Group at Lawrence Livermore National Laboratory

Education and Training

• Ohio State University	Electrical Engineering	M.S., 1989
• Fenn College of Engineering	Electrical Engineering	B.S., 1987

Summary of Qualifications

Ken Masica has been an Electrical Engineer in the Systems Engineering Group at Lawrence Livermore National Laboratory (LLNL) specializing in the areas of secure communication system design, network security architecture development, and cyber security vulnerability assessment. Twenty years' experience developing and deploying network and security solutions to support control and monitoring systems, building automation systems, advanced electrical metering infrastructures, VSAT satellite communications installations, wireless sensor networks, mission-critical emergency response communication systems, and national and international wide-area data networks. Have lead cyber security vulnerability assessments at numerous utility companies and federal agencies as part of the national critical infrastructure protection program for over ten years. Possess knowledge, skills, and experience related to assessing large-scale networking and computing infrastructures and control system architectures to identify potential vulnerabilities and harden them against cyber attack.

Professional Experience

- 1989 – Present
  Electrical Engineer, Systems Engineering Group at LLNL, Livermore California
- 1988 – 1989
  Network Programmer at The Ohio Supercomputer Center, Columbus Ohio
- 1985 – 1986
  Engineering Intern at NASA Glenn Research Center, Cleveland Ohio

Publications (selected)

K Masica, "How to Secure Your Power System", cover article for Electricity Today Magazine, June 2012.

K Masica, "Secure Your AMI", Electricity Today Magazine, March 2012 (AMI = Advance Metering Infrastructure)

K Masica, "Recommended Practices Guide for Securing Zig Bee Wireless for Industrial and Process Control Environments", for DHS US CERT Control System Security Program, April 2007, UCRL-TR-234688.

K Masica, "Recommended Practices Guide for Securing WLANs using 802.11i", for DHS US CERT Control System Security Program, October 2006, UCRL-TR-225541.

K Masica, "Analysis of Emerging Mote Technology as a Distributed Sensor Platform", LLNL Engineering Technology Base Program, April 2005, UCRL-TR-209922.

K Masica, "Assess Network Security with Nmap", E -Business Advisor magazine, July 2002.

K Masica, "Can Your Network Withstand an Attack?", Cover Article in Business Security Advisor magazine, December 2001.

K Masica, "Understand ing VPN and PKI Integration", Internet Security Advisor magazine, February 2001.

K Masica, "Understanding the IP Security Protocol", Internet Security Advisor magazine, October 2000.

K Masica, "Building Layer -3 Extranets with IPsec", Internet Security Advisor magazine, Nov/Dec 1999.

Synergistic Activities (selected)

- Member of the International Information Systems Security Certification Consortium, Inc., (ISC)^2 (www.isc2.org)
- Certified by (ISC)^2 as a Certified Information System Security Professional (CISSP)

Jessica Smith, Co-Principal Investigator

Education

Ph.D. Computer Science, Washington State University, est. 2015
"Hardware Reverse Engineering for Detection of Supply Chain Tampering"

M.S. Computer Engineering,University of Idaho, 2010
"An Investigation of Hardware Security in Multicore Architectures"

B.S. Computer Engineering, University of Idaho, 2009

Experience

Hardware Security Testing

- Performed hands-on analysis and testing of a variety of hardware compon ents, including processor architectures, COTS networking equipment, and SCADA control systems.

Security Research

- Performed research into hardware/software interactions and how those interactions affect security.
- Explored IC supply chain tampering threats and mitigations.
- Worked to develop advancements in detection of threats.

Cyber Security Analysis

- Analyzed network traffic and network components for security against outside attackers.
- Analyzed network traffic to extract information about flow and content.
- Performed malware reverse engineering.

Hardware/Software Design

- Designed and implemented algorithms in VHDL for instantiation on an FPGA.
- Created embedded programs to run on a range of embedded microprocessors and microcontrollers.

Relevant Employment History

Research Engineer 2011-Present

Secure Cyber Systems, National Security Directorate, PNNL, Richland WA.

- Performed penetration testing of SCADA power systems equipment.
- Led a small team in research of hardware IC reverse engineering algorithms and implementations.
- Assisted development of multi-use computing testbed.
- Performed analysis of networking equipment for security and functionality.
- Led a small team in the devel opment of software to detect specific behavior patterns in networked computers.

InfoSec Engineer/Scientist 2010-2011

Center for Integrated Intelligence Systems, National Security Engineering Center, MITRE, McLean VA.

- Researched software randomization at the assembly level and hardware randomization in FPGAs.
- Explored methods of implementing hardware security in FPGA based soft-core processors.
- Completed malware analysis tool evaluation.
- Analyzed computing assets, looking for possible illicit software or hardware modifications.

Research Assistant 2008-2010

Computer Science Dept., University of Idaho, Moscow, ID.

- Performed exploration Intel Nehalem and STI Cell Broadband Engine processor architectures.
- Designed and created experiments to test functionality and security of these processors.
- Guided other students in related research.
- Created curriculum for and led electronics laboratory sessions.

Publications

J. Smith, X. He, J. Alves-Foss. A Security Review of the Cell Broadband Engine Processor, Hawaii International Conference on System Sciences, 2010. Presenting Author.

R. Bradetich, P. Oman, J. Alves-Foss, **J. Smith**. Towards Resilient Multicore Architectures for Real-Time Controls, International Symposium on Resilient Control Systems, 2010.

Inventions

A Multi-Factor Authentication Method. Provisional Patent Application 61/371,059. 2010

8 Commitment Letters

8.1 LLNL

June 27, 2012

Dr. David Manz
Pacific Northwest National Laboratory
Secure Cyber Systems Group
902 Battelle Boulevard
P.O. Box 999 MSIN J4-45
Richland, WA 99352

RE: Letter of Support – Cybersecurity for Energy Delivery Systems Research Call RC-CEDS-2012-012 Project 3 - Detect Compromise of Supply Chain Integrity

Dear Dr. Manz:

This letter is to indicate the support of Lawrence Livermore National Laboratory (LLNL) for Pacific Northwest National Laboratory's proposal to detect and address compromise of supply chain integrity, titled "SCI-FI: Supply Chain Integration For Integrity."

LLNL is a multi-disciplinary national research laboratory located in Livermore, California with a mission of strengthening the security of the United States through development and application of world-class science and technology. More information can be obtained online at www.llnl.gov or by contacting Public Affairs at the number and address provided below.

We are looking forward to collaborating with PNNL on this upcoming project.

Sincerely,

Robert [illegible]

Address and Numbers
Lawrence Livermore National Laboratory
7000 East Ave., Livermore, CA 94550-9234 (Deliveries)
P.O. Box 808, Livermore, CA 94551-0808 (Mail)

Main Operator (925) 422-1100
Fax (925) 422-1370, Fax verification (925) 422-1100

Information Line
The Public Affairs information line is available to answer questions about the Laboratory's mission, programs, and activities and can be reached at (925) 422-4599.

8.2 ORNL

OAK RIDGE NATIONAL LABORATORY

MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

P.O. Box 2008
Oak Ridge, TN 37831-6253
Phone: (865) 574-4333
Fax: (865) 574-9869
E-mail: kellerm@ornl.gov

July 2, 2012

Dr. David Manz
Pacific Northwest National Laboratory
Secure Cyber Systems Group
902 Battelle Boulevard
P.O. Box 999 MSIN J4-45
Richland, Washington 99352

Dear Dr. Manz:

Letter of Support – Cybersecurity for Energy Delivery Systems Research Call RC-CEDS-2012-012, Project 3 - Detect Compromise of Supply Chain Integrity

This letter is to indicate the support of Dr. Josef D. Allen of the Oak Ridge National Laboratory (ORNL) for Pacific Northwest National Laboratory's (PNNL) proposal to detect and address compromise of supply chain integrity entitled "Holistic Supply Chain Analysis."

ORNL is the largest science and energy national laboratory in the Department of Energy system. ORNL's scientific programs focus on materials, neutron science, energy, high-performance computing, systems biology and national security.

ORNL partners with the state of Tennessee, universities, and industries to solve challenges in energy, advanced materials, manufacturing, security, and physics. The laboratory's science and technology innovations are translated into applications that add value throughout the world.

Specifically for the Cybersecurity for Energy Delivery Systems call Dr. Allen will support trusted supply chain integrity. Dr. Allen is the lead scientist for Cyber-Physical Security for Critical Infrastructure Protection and Security for the Energy and Environmental Sciences Directorate. His duties include Systems' Security Engineering, Secure TCP/IP for layers 3, 4 and 5, and Secure Network Architecture for SCADA and Substation Automation. He serves on the Department of Energy's Office of Electricity Cyber Security for Electric Delivery Systems. He is also a member of the National SCADA Test Bed, which is comprised of several national laboratories. Dr. Allen is our lead as it pertains to cutting edge research cyber-physical systems and devices.

Dr. Allen is also a part of the Trusted Computing Group (TCG). TCG is a not-for-profit organization formed to develop, define, and promote open, vendor-neutral, industry standards for trusted computing building blocks and software interfaces across multiple platforms.
We are looking forward to collaborating with PNNL on this upcoming project. If I can be of further assistance please do not hesitate to contact me.

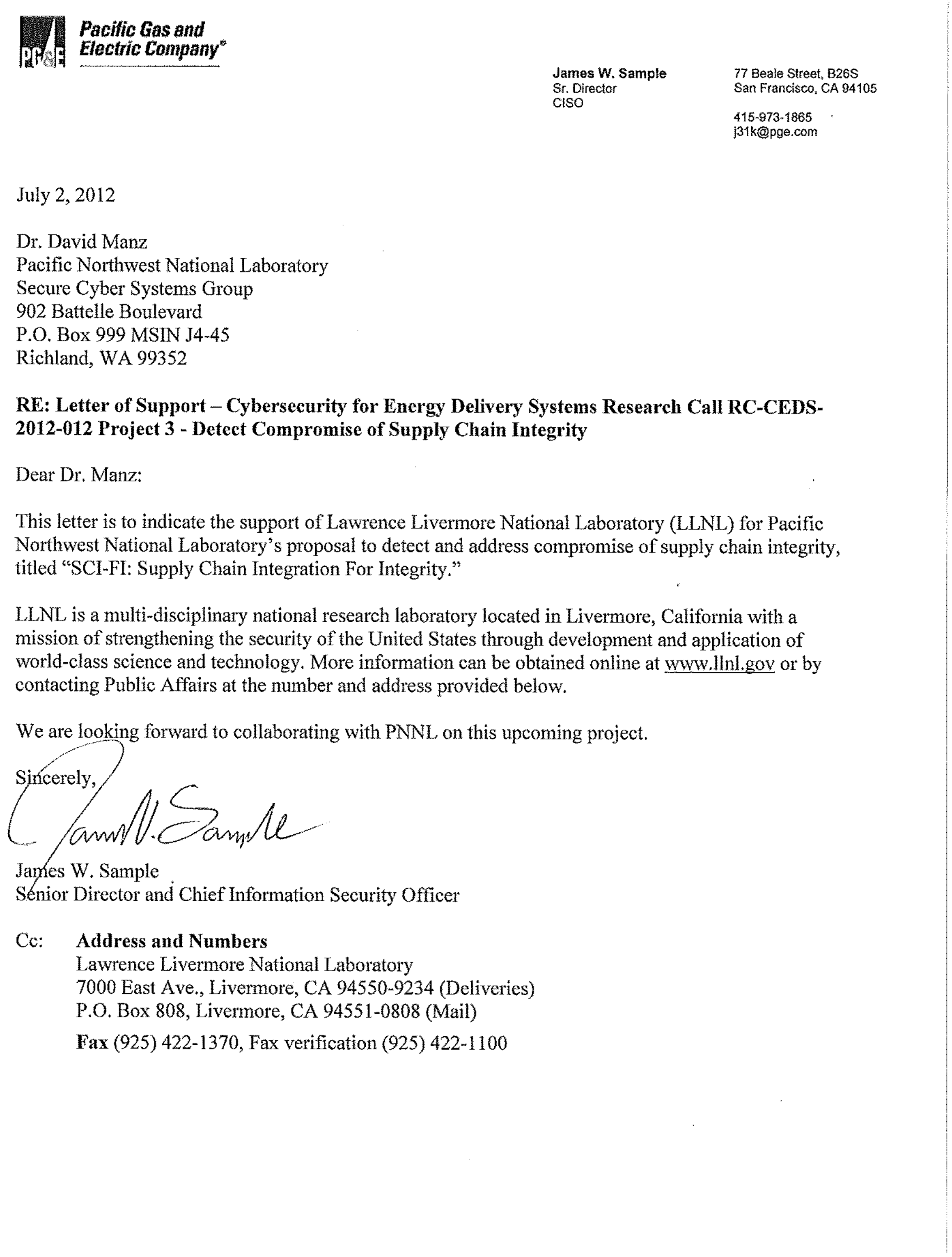
Sincerely,

Martin Keller

Martin Keller, Ph.D.
Associate Laboratory Director
Energy and Environmental Sciences Directorate

MK:kjm

8.3 Pacific Gas and Electric



Pacific Gas and Electric Company®

James W. Sample
Sr. Director
CISO

77 Beale Street, B26S
San Francisco, CA 94105

415-973-1865
j31k@pge.com

July 2, 2012

Dr. David Manz
Pacific Northwest National Laboratory
Secure Cyber Systems Group
902 Battelle Boulevard
P.O. Box 999 MSIN J4-45
Richland, WA 99352

RE: Letter of Support – Cybersecurity for Energy Delivery Systems Research Call RC-CEDS-2012-012 Project 3 - Detect Compromise of Supply Chain Integrity

Dear Dr. Manz:

This letter is to indicate the support of Lawrence Livermore National Laboratory (LLNL) for Pacific Northwest National Laboratory's proposal to detect and address compromise of supply chain integrity, titled "SCI-FI: Supply Chain Integration For Integrity."

LLNL is a multi-disciplinary national research laboratory located in Livermore, California with a mission of strengthening the security of the United States through development and application of world-class science and technology. More information can be obtained online at www.llnl.gov or by contacting Public Affairs at the number and address provided below.

We are looking forward to collaborating with PNNL on this upcoming project.

Sincerely,

James W. Sample
Senior Director and Chief Information Security Officer

Cc: **Address and Numbers**
Lawrence Livermore National Laboratory
7000 East Ave., Livermore, CA 94550-9234 (Deliveries)
P.O. Box 808, Livermore, CA 94551-0808 (Mail)
Fax (925) 422-1370, Fax verification (925) 422-1100

8.4 DMI

DMI *Enterprise Transformation
6550 Rock Spring Drive *7th Floor *Bethesda *MD 20817
240.223.4800 *fax 240.223.4888 *DMInc.com

DMI

July 2, 2012

Dr. David Manz
Pacific Northwest National Laboratory
Secure Cyber Systems Group
902 Battelle Boulevard
P.O. Box 999 MSIN J4-45
Richland, WA 99352

RE: Letter of Support*Cybersecurity for Energy Delivery Systems Research CallCEDS-2012-012 Project 3 - Detect Compromise of Supply Chain Integrity

Dear Dr. Manz:

This letter is to indicate the support of Digital Management, Inc. (DMI) for Pacific Northwest National Labo****** * * * ******** * ** * ****** * *** * ise of supply chain integrity, ****** * * ******** * ****** *
***** * ******** * * ******* * ***** * * *

DMI is a leading IT solutions and business strategy consulting firm focused on providing solutions that transform enterprise operations in government and business by dependably bridging the gap between business strategy and mission success. Leveraging technology as an efficient, economical means to an end, DMI crafts solutions that result in increasingly interoperable, responsive, and cost-effective
*********** * * *** * * * ********** * ** * ******** * ******** * *** * ********* * * ********* * ********* *
Transformation, Trusted Computing, Supply Chain Improvement, Software Systems Modernization, Enterprise Information Management, Cybersecurity, and Healthcare IT, has resulted in dramatic revenue growth, and a growing client base that now includes fourteen of the fifteen U.S. Federal Departments. DMI is headquartered in Bethesda MD, with satellite and project offices throughout the world.

With regard to this proposal, DMI brings specific expertise in creating industry-driven security solutions to the Department of Defense and federal civilian agencies, with a focus on trusted computing solutions. DMI brings the highest level of expertise in trusted computing available in the market today. DMI provides trusted computing strategy, architecture, design and research services aimed at building trusted computing into the hardware of all manner of platforms. Our trusted computing customers includ e the Department of Defense, the U.S. Air Force, large companies in the defense industrial base, and industry leading companies in the high technology hardware and software industries. DMI is also extremely active in the Trusted Computing Group (TCG), the international standards body focused on trusted computing. DMI participates in a half dozen work groups, including holding the chair of the TPM Work Group. The TPM Work Group writes the specification for the Trusted Platform Module, an important example o f a Trust Anchor.

DMI also brings domain experience around improving the security and performance of the supply chain. DM* * * * ****** * ***** * ******** * **** * *****gencies and other large organizations improve performance, compliance and security across the various organizations that make up their supply chains. DMI has written white papers on supply chain security and has teamed with GXS, the global leader in B2B integration and supply chain automation solutions, offering strategic and tactical consulting, security enhancements, supply chain visibility solutions and process automation solutions.

Addressing challenges around protecting our supply chain is critical to ensuring reliability of our systems and defending our national security. Applying industry experience and working with a strong, interdisciplinary team to incorporate trusted computing capabilities for supply chain integrity can make a significant difference in enabling us to meet these challenges. We are looking forward to col laborating with PNNL on this upcoming project.

Sincerely,

Andrew J Musliner

Andrew J. Musliner
Executive Vice President
Chief Technology & Innovation Officer