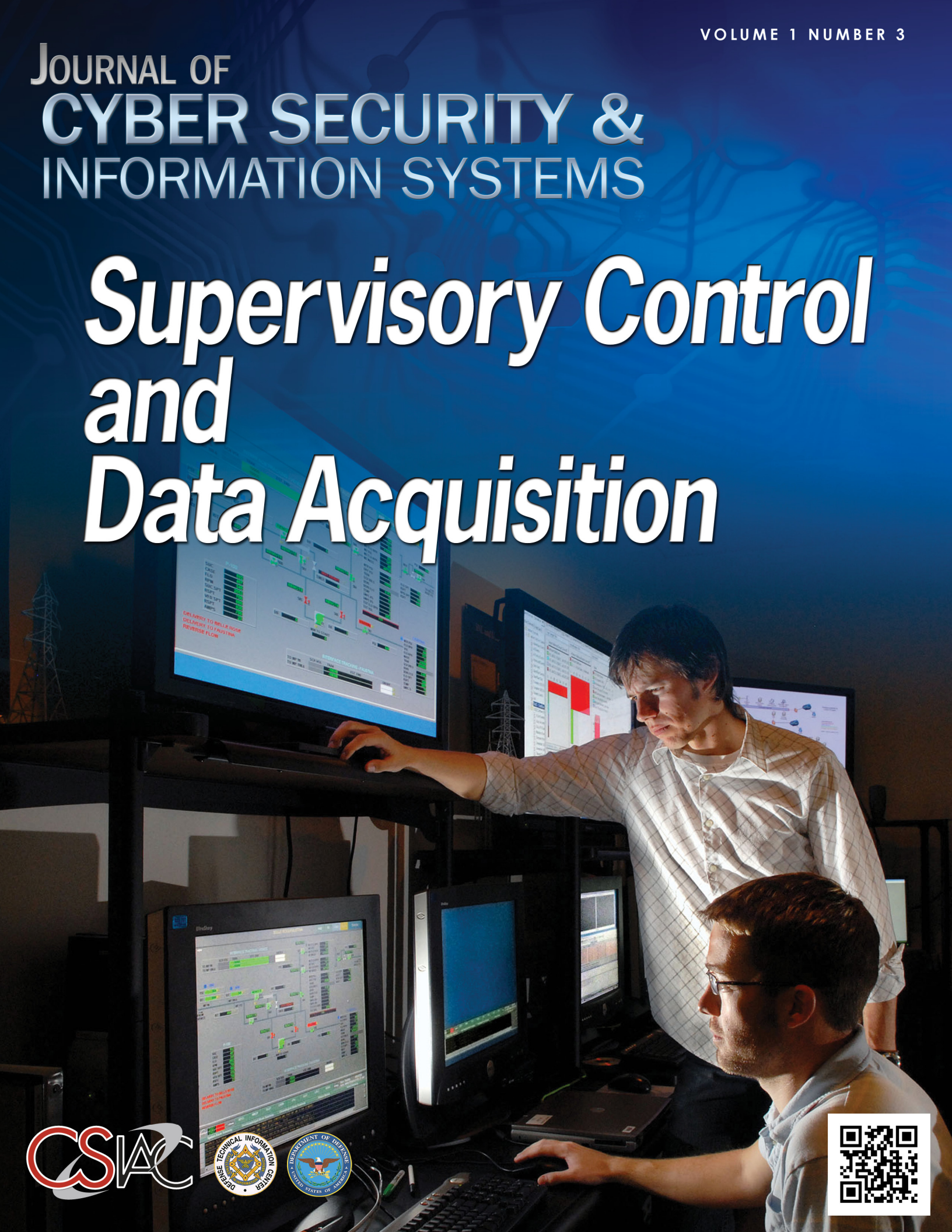
VOLUME 1 NUMBER 3

JOURNAL OF
CYBER SECURITY &
INFORMATION SYSTEMS

Supervisory Control and Data Acquisition



The Efficacy and Challenges of SCADA and Smart Grid Integration

By Les Cardwell and Annie Shebanow

The advent and evolution of the Smart Grid initiative to improve the electric utility power infrastructure has brought with it a number of opportunities for improving efficiencies, but along with those benefits come challenges in the effort to assure safety, security, and reliability for utilities and consumers alike. One of the considerations in designing the capabilities of the Smart Grid is the integration of Supervisory Control and Data Acquisition (SCADA) systems to allow the utility to remotely monitor and control network devices as a means of achieving reliability and demand efficiencies for the utility as a whole. Given the ability of these systems to control the flow of electricity throughout the network, additional planning and forethought is required to ensure all possible measures for preventing compromise are considered. This work discusses the overall architecture(s) used today and some of the measures currently implemented to secure those architectures as they evolve. More importantly, it considers simplifying the complexity of implementing the many standards put forth by applicable standards and regulatory bodies as a means to achieve realistic governance.

Introduction

Utility infrastructures represent privileged targets for cyber terrorists or foreign state-sponsored hackers. There are a number of challenges to achieve a base-level security across the utility spectrum. The challenges are due to limited budgets, privately owned control systems in utility infrastructures, and the complexity in decomposing the myriad sets of requirements from competing regulatory bodies each with their own frameworks. The process of developing a functional, secure infrastructure requires technology skills and understanding how and why all applied technologies interact with each other.

In this section, the SCADA and smart grid are explained to discuss the efficacy and challenges in the integration process.

SCADA

Supervisory Control and Data Acquisition (SCADA) systems are basically Process Control Systems (PCS) that are used for monitoring, gathering, and analyzing real-time environmental data from a simple office building or a complex nuclear power plant. PCSs are designed to automate electronic systems based on a predetermined set of conditions, such as traffic control or power grid management. Some PCSs consist of one or more remote terminal units (RTUs) and/or Programmable Logic Controllers (PLC) connected to any number of actuators and sensors, which relay data to a master data collective device for analysis. Gervasi (2010) described SCADA systems with the following components:

1. *Operating equipment:* pumps, valves, conveyors, and substation breakers that can be controlled by energizing actuators or relays.

2. *Local processors:* communicate with the site's instruments and operating equipment. This includes the Programmable Logic Controller (PLC), Remote Terminal Unit (RTU), Intelligent Electronic Device (IED), and Process Automation Controller (PAC). A single local processor may be responsible for dozens of inputs from instruments and outputs to operating equipment.

3. *Instruments*: in the field or in a facility that sense conditions such as pH, temperature, pressure, power level, and flow rate.

4. *Short-range communications*: between local processors, instruments, and operating equipment. These relatively short cables or wireless connections carry analog and discrete signals using electrical characteristics such as voltage and current, or using other established industrial communications protocols.

5. *Long-range communications:* between local processors and host computers. This communication typically covers miles using methods such as leased phone lines, satellite, microwave, frame relay, and cellular packet data.

6. *Host computers*: act as the central point of monitoring and control. The host computer is where a human operator can supervise the process, as well as receive alarms, review data, and exercise control.

Figure 1 displays a high-level overview of SCADA architecture, where the Remote Stations might be an Electric Substation, the SCADA network on one network segment, with other organization network on differing network segments. With advancements in the computing field, the integration of digital electronics devices play an important role in the manufacturing industry, wherein manufacturing plants utilize PLCs/RTUs to control the devices, and develop distributed and large complicated systems in which intelligent systems are part of the manufacturing control systems processes.

"Most often, a SCADA system will monitor and make slight changes to function optimally; SCADA systems are considered closed loop systems and run with relatively little human intervention. One of the key processes of SCADA is the ability to monitor an entire system in real time. This is facilitated by data acquisitions including meter reading, checking statuses of sensors, etc. that are communicated at regular intervals depending on the system" (Abawajy & Robles, 2010).

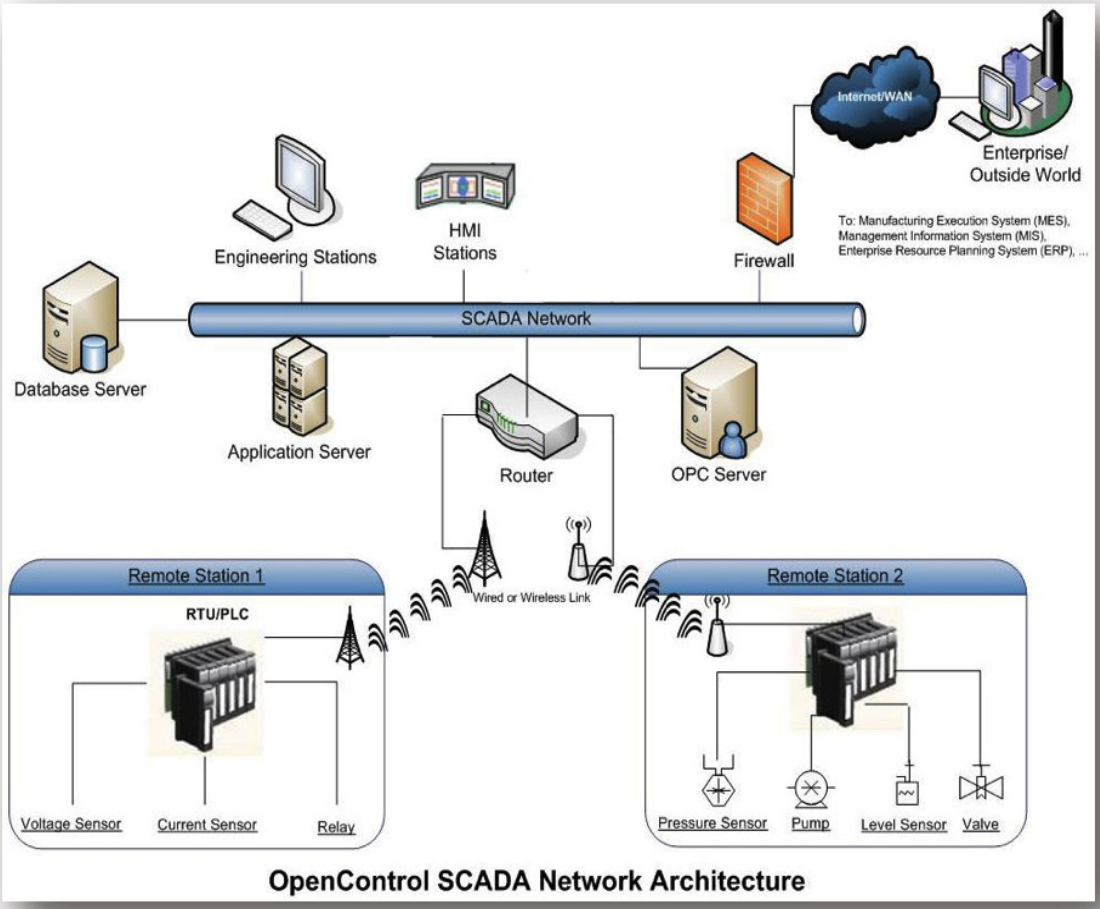


Figure 1: SCADA Network (Source: www.buraq.com)

Smart Grid

The Smart Grid domain is comprised of and concerned with distributed intelligence including data decentralization, distributed generation and storage, and distribution system automation and optimization. Customer involvement and interaction is a consideration, as are micro-grids, and high-consumption electric devices including plug- in hybrid electric vehicles (PHEV) (Collier, 2010).

The Smart Grid is by definition about real-time data and active grid management via fast two-way digital communications through the application of technological solutions to the electricity delivery infrastructure. Connectivity exists between (and within) the electric utility, utility's devices, consumer devices (In Home Devices, or IHDs), and third-party entities either as vendors, consumers, or regulatory bodies. Smart Grid includes an intelligent monitoring system that tracks the flow of electricity throughout the electrical network, and incorporates the use of superconductive transmission lines to manage power fluctuations, loss, and co-generation integration from solar and wind.

At its most efficient, the Smart Grid can control in-home devices that are non-critical during peak power usage-times to reduce demand, and return their function during non-peak hours. Proposals for optimization include smart electric grid, smart power grid, intelligent grid (or intelligrid), Future Grid, and the more modern intergrid and intragrid. In addition to leveling (or normalizing) electric demand, the ability to manage consumption peaks can assist in avoiding brown-outs and black-outs when demand exceeds supply, and allow for maintaining critical systems and devices under such conditions (Clark & Pavloski, 2010).

Figure 2 displays a high-level communication flow between different components in a Smart Grid.

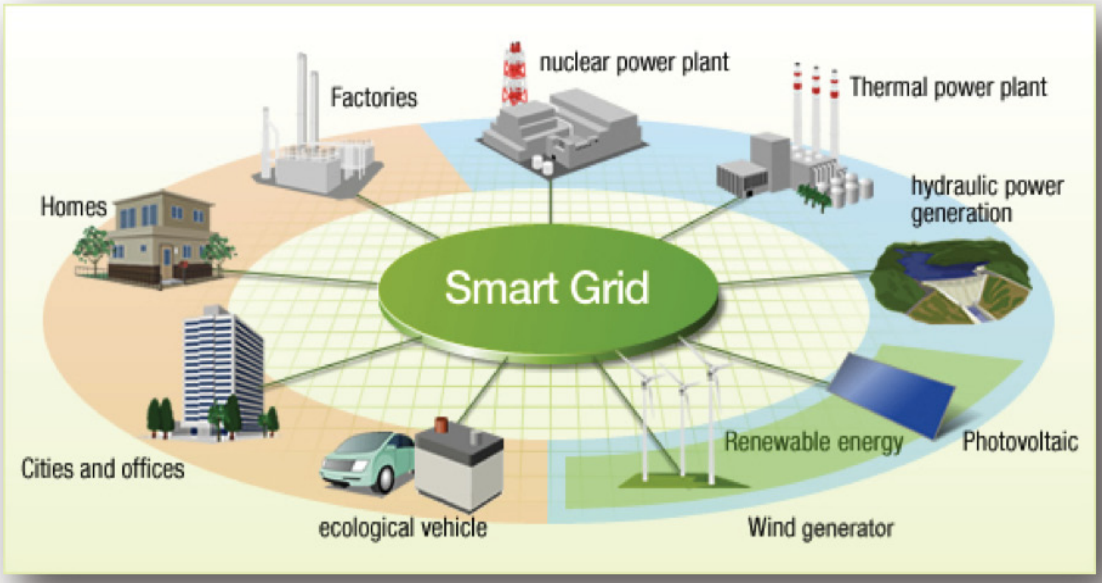


Figure 2: Smart Grid (Source: http://www.consoglobe.com)

The Smart Grid initiative has spawned a significant movement toward the modernization and evolution of the electric utility infrastructure, and aims to bring it into today's advanced communication age both in function and in architecture. That evolution brings with it a number of organizational, technical, socio-economic, and cyber security challenges. The breadth and depth of those challenges is not trivial, and a number of regulatory bodies have taken up the initiative to bring their own requirements into alignment with these new challenges. The initiative has also offered many opportunities for researchers, scientists, and enterprise architects to advance the state of security assurance; it also affords technologists the opportunity to explore new areas for exploiting means of data communication among distributed and remote networks and their devices.

Smart Grid / SCADA Integration

SCADA integration into the Smart Grid is not difficult, and connected by both electrical and data networks, allows for central and distributed aggregation of information and control over the entire utility electrical device network as depicted in Figure. 3.

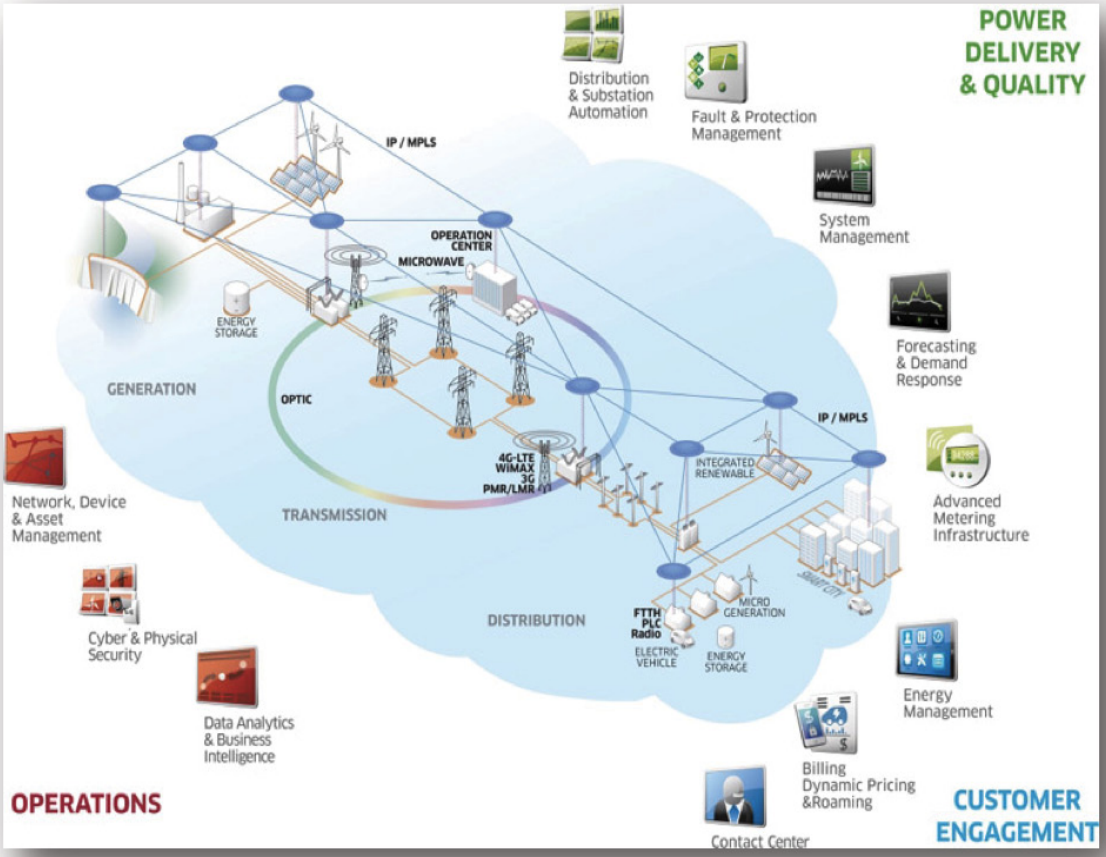


Figure 3: SCADA/Smart Grid integration (Source: http://www2.alcatel-lucent.com)

SCADA empowers the consumer by interconnecting energy management systems to enable the customer to manage their own energy use and control costs. It allows the grid to be self-healing by instantly responding automatically to outages, power quality issues, and system problems. Properly configured, it is tolerant to attack—both physical and cyber-attacks—and optimizes the grid assets by monitoring and optimizing those assets while minimizing maintenance and operations costs. Further, it also enables competitive energy markets and mitigates the bloat often incurred in the effort to obtain pricing guarantees.

To adequately deliver and administer the products and services made possible by the Smart Grid, intelligence and control need to exist along the entire supply chain. This includes the generation and transmission of electricity from inception to delivery end-points at the customer's side of the meter, and includes both fixed and mobile devices in the architecture.

Digital communications on a Smart Grid occur over a variety of devices, technologies, and protocols that include wired and wireless telephone, voice and data dispatch radio, fiber optics, power line carriers, and satellite. Decision Control Software (DCS) allows for dynamic grid management that involves monitoring a significant number of control points. To be fully effective and operational, monitoring occurs for every power line and piece of equipment in the distribution system, in addition to allowing the customers to monitor and control their own devices and usage. This results in considerable volumes of data to be organized, analyzed, and used for both manual and automated decision software that comes in two basic categories: decentralized and back office.

Decentralized software is necessary due to the magnitude of the devices and data collection and computation, which precludes a centralized data collection solution. As the technology matures, intelligent electronic devices (IEDs) will evolve to mitigate the collection, organization, and data analysis necessary for performing data routing, decision making, and other actions that may be necessary based on the information received. This functionality exists either as part of the firmware, or via configurable functions and settings within each device.

Back office software is typically that software which is used as part of the utility's line-of-business (LOB) software solutions necessary to conduct the business of the organization. This typically includes, but is not limited to the following:

- Accounting & Business Systems (ABSs)
- Customer Information Systems (CISs)
  - Customer Billing & Payment
  - Customer Relationship Management (CRM)
- Work & Workforce Management
  - Performance & Productivity Management
- Engineering & Operations (E&Os)
- Engineering Analysis
  - Circuit Modeling & Analysis
  - Reliability Analysis
- Real-Time Distribution Analysis
  - Outage Management System (OMS)
  - Active Distribution Grid Management
- Geographic Information Systems (GISs)
- Interactive Voice Response (IVR)

The net effect on these solutions by the deployment of IEDs and two-way digital communications is a more powerful, useful, and effective solution set for both the utility and the consumer.

SCADA and Smart Grid Security Considerations

Hentea (2008) discusses the evolution and security issue escalation of SCADA and the Smart Grid due in large part to the advent of the internet and rise in terrorist threats. Additionally, the introduction of new protocols, LAN/WAN architectures, and new technologies such as encryption and information assurance applications on the shared network(s) raise new sets of security concerns.

The increased functionality of SCADA and the Smart Grid architecture leads to control systems that are escalating in complexity and have become time critical, embedded, fault tolerant, distributed, intelligent, large, open sourced, and heterogeneous, all which pose their own program vulnerabilities. Ranked high on the list of government concerns are threats against SCADA systems. Unfortunately, mostly due to the complexities involved and resources required, the threats are too often trivialized and most organizations are slow to implement enhanced security measures to combat these threats. Key requirement areas for addressing these threats are critical path protection, strong safety policies, procedures, knowledge management, and system development skills that place security architecture at the forefront of requirements.

In considering potential risks in the act of collecting data from distributed access points using wireless radio frequency technology, "The very nature of Radio Frequency (RF) technology makes Wireless LANs (WLANs) open to a variety of unique attacks. Most of these RF-related attacks begin as exploits of Layer 1 (Physical – PHY) and Layer 2 (Media Access Control – MAC) of the 802.11 specification, and then build into a wide array of more advanced assaults, including Denial of Service (DOS) attacks. In Intelligent Jamming, the jammer jams physical layer of WLAN by generating continuous high power noise in the vicinity of wireless receiver nodes" (Jha, Kumar & Dalal, 2010).

To combat some of these risks, Teixeira, Dán, Sandberg, and Johansson (2010) discuss the need for the use of litmus and metrics in the form of state estimators commonly used in power networks to detect problems and optimize power flows. These are usually located in central control centers and receive significant data measurements sent over unencrypted communication channels, making cyber security an important issue. Bad data detection (BDD) schemes exist as energy management systems (EMSs) state estimation algorithms to

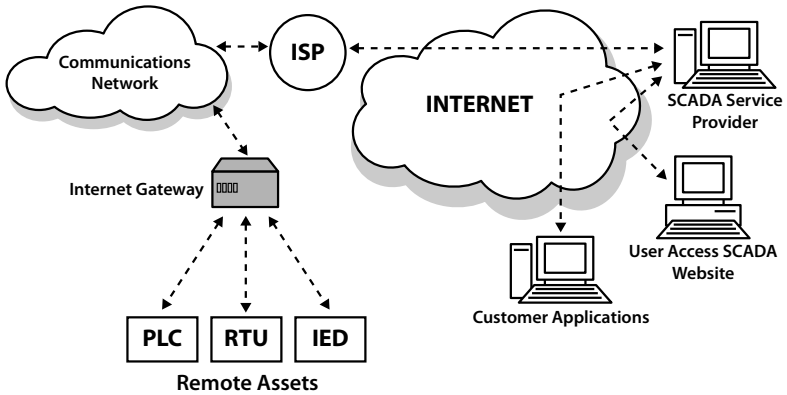


Figure 4 – Internet SCADA Architecture (Source: Gervasi, 2010)

detect outliers and inconsistencies in the data, and are based on high measurement redundancy. While these methods may detect a basic cyber-attack, additional security considerations should be implemented to deter an intelligent attacker intent on gaining access and control of a SCADA system directly or through one of the Smart Grid devices.

Integration into the Internet Figure. 4 provides a delivery medium available to most consumers, and can provide advantages in the form of control, distribution, and communication. The Internet utilizes Hybrid fiber-cable (HFC), digital subscriber line (DSL), broadband over power lines (BPL), wireless (Wi-Fi and WiMAX), fiber, satellite, and utilizes wholly-owned and operated networks and third-party networks where feasible and cost effective.

SCADA also creates a number of additional security issues since the electrical power network is a critical infrastructure. Without Internet connectivity, SCADA already contends with security issues, and additional methods of penetration via the internet make it more vulnerable. There are a number of common security issues with SCADA:

- A lack of concern about security and authentication in the design, deployment, and operation of existing Control System networks
- The belief that SCADA systems have the benefit of security through obscurity, through the use of specialized protocols and proprietary interfaces
- The belief that SCADA networks are secure because they are purportedly physically secured
- The belief that SCADA networks are secure because they are supposedly disconnected from the Internet
- IP Performance Overhead of Control Systems connected to the Internet

Among the suggestions to further enhance SCADA and Internet security, Gervasi (2010) offers a "Crossed-Crypto Scheme" for securing communications. "There are major types of encryptions in cryptography: the symmetric encryption and the asymmetric encryption. From the two major types of encryptions we can say that Asymmetric encryption provides more functionality than symmetric encryption, at the expense of speed and hardware cost." The scheme integrates into the communication of the SCADA master and SCADA assets wherein the plain text data transmits using the AES algorithm for encryption, then encrypts the AES key using ECC. The cipher text of the message and the cipher text of the key are then sent to the SCADA assets, also encrypted using ECC techniques. "The cipher text of the message digest is decrypted using ECC technique to obtain the message digest sent by the SCADA Master. This value is compared with the computed message digest. If both of them are equal, the message is accepted; otherwise it is rejected."

Chauvenet and others also consider enhancements to the communication stack for power-line communication (PLC) based on and the adaption of the IEEE802.15.4 standard protocol, which is constrained by the low-power, lossy, and low data-rate context of power-line transceiver using pulse modulation, using open standards using IPv6 at the network level with the 6LoWPAN adaption (Chauvenet, Tourancheau, Genon-Catalot, Goudet, and Pouillot,2010). In their paper, they posit that "this allows for a full network layer stack and results in efficient routing in our low power, low data-rate and lossy network context" and cross compare their posit with other available communication solutions.

Other standards and maturity models are being developed to address the growing security concerns for the evolving energy distribution models (Fries, Hof, & Seewald, 2010) such as security enhancements to the IEC61850, which is a standardized communication services and standardized data model for communication in energy automation. Therein lies the challenge. The number of standards, recommendations, requirements, and frameworks that are evolving in the attempt to address the growing security challenges for securing SCADA and the Smart Grid is not trivial. Further, each utility, depending on the services the utility provides, are subject to many of these standards, each prescribing recommendations that are redundant across standards. Wading through multiple sources of these in an effort to be thorough is daunting, resource intensive, and a moving target that requires policies and procedures to ensure all recommendations are vetted against both existing assets and any new assets. Ensuring that the risks, many as unknown and potentially pervasive, are not trivialized and rationalized away is a challenge.

Security Integration Improvement – Addressing Cybersecurity Risks

A posit by Langner and Pederson (2013) suggests that putting emphasis on establishing frameworks for risk management, and relying on voluntary participation of the private sector that owns and operates the majority of US critical infrastructure are together a recipe for continued failure. The reason for this is the reliance on the concept of risk management framed as a problem in business logic, which ultimately allows the private sector to argue the hypothetical risk away. The authors suggest that a policy-based approach

(vs. a risk-assessment based approach) that sets clear guidelines would avoid perpetuating the problem. They also argue the distinction between a critical and a non-critical systems only contradicts pervasiveness and sustainability of the effort in arriving at robust and well-protected systems.

As was recently asserted by Cardwell (2013) in response to the National Institute of Standards and Technology's (NIST) "RFI – Framework for Reducing Cyber Risks to Critical Infrastructure" driven by the recent Executive Order "Improving Critical Infrastructure Cybersecurity" (NIST, 2013), the "…issue is the 'expanding redundant complexity' of the current approach to the problem domain. While one can appreciate the efforts in gathering more information from the industry at large for establishing and improving frameworks to raise the overall level of cybersecurity across the utility industry, the problem is that it does not address the inherent complexity of the problem. It only exacerbates it by creating yet more administrative requirements for decomposing and resolving the problem domain for each utility."

Rather than asking every utility to wade through every applicable (to that utility) standard, recommendation, and framework, the assertion suggests that a "single-source" methodology that eliminates redundancy across all frameworks be adopted and provided for addressing the complexity and achieving a Digital Systems Security (DSS) Cybersecurity standard across the US Utility spectrum. Using a single-source tool as litmus, the outcome is a reduction in administrative and redundant efforts otherwise required to manage the information between multiple systems, and serves as a living digital document of the DSS domain, thus simplifying the process further.

One such tool does currently exist: the Cyber Security Evaluation Tool (CSET) (DHS, 2011) by the Department of Homeland Security (DHS), although improvements are still being applied to improve its efficacy. Even with such an application, while the process is certainly not "easy" for any utility, it is relatively simple in comparison to wading through all the various requirements and recommendations, hoping to achieve a full decomposition of each. Simplifying the DSS Cybersecurity process in this fashion will save utilities—both individually and collectively—significant amounts of time, and resources, and could galvanize the DSS efforts for both the regulatory bodies and the utility industry combined.

	Manage Cybersecurity Risk	Common Practices
MIL1	a. Cybersecurity risks are identified b. Identified risks are mitigated, accepted, tolerated, or transferred	1. Initial practices are performed but may be ad hoc
MIL2	c. Risk assessments are performed to identify risks in accordance with the risk management strategy d. Identified risks are documented e. Identified risks are analyzed to prioritize response activities in accordance with the risk management strategy f. Identified risks are monitored in accordance with the risk management strategy g. A network (IT and/or OT) architecture is used to support risk analysis	1. Practices are documented 2. Stakeholders of the practice are identified and involved 3. Adequate resources are provided to support the process (people, funding, and tools 4. Standards and/or guidelines have been identified to guide the implementation of the practices
MIL3	h. The risk management program defines and operates risk management policies and procedures that implement the risk management strategy i. A current cybersecurity architecture is used to support risk analysis j. A risk register (a structured repository of identified risks) is used to support risk management	1. Activities are guided by policies (or other organizational directives) and governance 2. Activities are periodically reviewed to ensure they conform to policy 3. Responsibility and authority for performing the practice is clearly assigned to personnel 4. Personnel performing the practice have adequate skills and knowledge

While establishing such a tool as litmus for evaluating the level of DSS maturity for a given utility, some additional thought went into the subject using the Capability Maturity Model Integration (CMMI Institute, 2010) to assist utilities in that effort. That effort resulted in a modified CMMI model labeled as the Electricity Subsector Cybersecurity Capability Maturity Model (DHS, 2012).

Electricity Subsector Cybersecurity Capability Maturity Model

Efforts in establishing standard security practices that can be broadly applied and implemented for the electric utility industry can be found in the evolving "Electricity Subsector Cybersecurity Capability Maturity Model (ES-C2M2), and is discussed by Balijepalli, Khaparde, Gupta, and Pradeep (2010) as a tool which "can guide the transformation of an entire power grid forward towards smarter grid. This will assess the utility grid state for moving towards the vision of Smart Grid. Some of the utilities are planning their Smart Grid road maps and investments using ES-C2M2. This helps to establish a shared picture of the Smart Grid journey, communicate the Smart Grid vision, and internally and externally assess current opportunities, choices, and desired levels. This also helps in the strategic and decision making framework to develop business, investment and rate cases, build an explicit plan to move from one level to another, measure progress using key performance indicators (KPIs), benchmark and learn from others." The ES-C2M2 parallels the CMMI model in form as follows, although the ES-C2M2 to date only measures through Level 3.

There are eight domains of logical groupings with related capabilities and characteristics at each maturity level as shown in Figure 6. Maturity Levels are defined for each domain to assess the current state of a utility's overall maturity level.

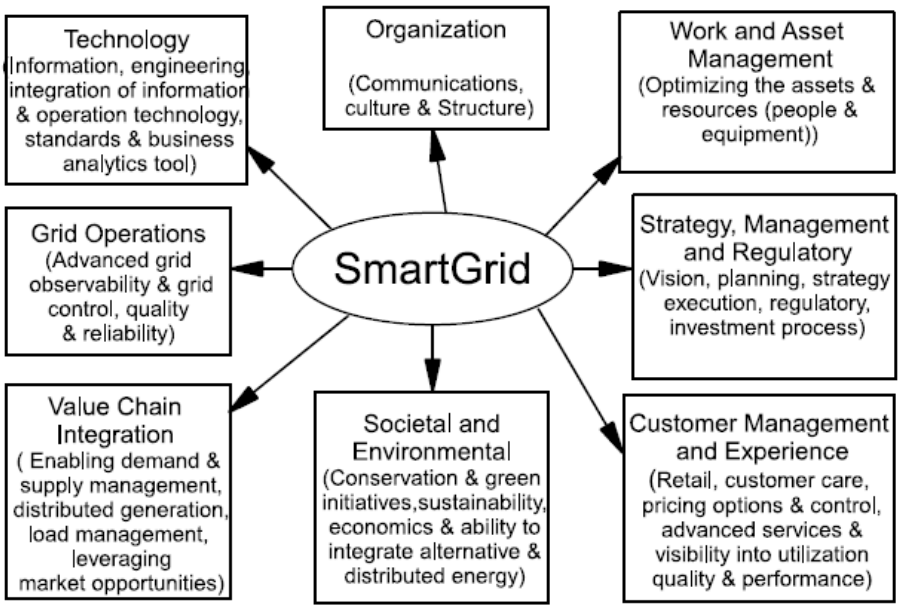


Figure 6 – Eight domains and Smart Grid elements (Source: DHS, 2011)

It has been suggested by Cardwell (2013) that the ES-C2M2 be used as litmus for helping utilities achieve and maintain a Maturity Level 3 status, though it is currently used simply as a tool for a utility to assess their own status.

Conclusion

In this paper, we explored the Smart Grid initiative and described integration of SCADA systems into the Smart Grid, including an overview of the problem domain as a whole. We then showed that the outer bounds and limits of the security requirements are as yet not known, and until the architecture and its implementation are complete, repeatable, and mature, the "wicked complexity" of systems will exist due to the "unknown" aspects of cybersecurity. Also discussed are possible approaches for addressing the complexities in securing a utility's cyber-structure, and some of the efforts that seek to address the security concerns and requirements of the initiative. While solutions are forthcoming, a pervasive industry-wide answer to the challenge is still evolving.

References

Abawajy, J., & Robles, R. J. (2010). Secured Communication Scheme for SCADA in Smart Grid Environment. *Journal of Security Engineering, 7*(6), 12.

Balijepalli, V. S. K. M., Khaparde, S., Gupta, R., & Pradeep, Y. (2010). *SmartGrid initiatives and power market in India*.

Cardwell, L. (2013, February 28). *Comments received in response to: Federal register notice developing a framework to improve critical infrastructure cybersecurity*. Retrieved on April 10 from http://csrc.nist.gov/cyberframework/rfi_comments/central_lincoln_pud_022813.pdf

Chauvenet, C., Tourancheau, B., Genon-Catalot, D., Goudet, P. E., & Pouillot, M. (2010). *A communication stack over PLC for multi physical layer IPv6 Networking*.

Clark, A., & Pavlovski, C. J. (2010). Wireless Networks for the Smart Energy Grid: Application Aware Networks. *Proceedings of the International MultiConference of Engineers and Computer Scientists, 2*.

CMMI Institute. (2010, November). Capability maturity model integration. Retrieved from http://cmmiinstitute.com/

Collier, S. E. (2010). Ten steps to a smarter grid. *Industry Applications Magazine, IEEE, 16*(2), 62-68.

DHS. (2011, January 24). Cyber security evaluation tool. Retrieved from http://ics-cert.us-cert.gov/satool.html

DHS. (2012, May 31). Electricity subsector cybersecurity capability maturity model. Retrieved from http://energy.gov/oe/services/cybersecurity/electricity-subsector-cybersecurity-capability-maturity-model

Fries, S., Hof, H. J., & Seewald, M. (2010). *Enhancing IEC 62351 to Improve Security for Energy Automation in Smart Grid Environments*.

Gervasi, O. (2010). Encryption scheme for secured Communication of web based control systems. *Journal of Security Engineering, 7*(6), 12.

Hentea, M. (2008). Improving security for SCADA control systems. *Interdisciplinary Journal of Information, Knowledge, and Management, 3*, 73-86.

Jha, R. K., Kumar, R. A., & Dalal, U. D. *Performance Comparison of Intelligent Jamming in RF (Physical) Layer with WLAN Ethernet Router and WLAN Ethernet Bridge*. Paper presented at the Proceedings of the 2010 International Conference on Advances in Communication, Network, and Computing.

Langner, R., & Pederson, P. (2013). Bound to fail: Why cyber security risk cannot simply be "managed" away. Retrieved on April 10 from http://www.whitehouse.gov/the-press-office/2013/02/12/executive-order-improving-critical-infrastructure-cybersecurity

NIST. (2013, February 12). Cybersecurity framework. Retrieved from http://www.nist.gov/itl/cyberframework.cfm

Teixeira, A., Dán, G., Sandberg, H., & Johansson, K. H. (2010). A cyber security study of a SCADA energy management system: Stealthy deception attacks on the state estimator. *Arxiv preprint arXiv:1011.1828*.

About the Authos(s)

Dr. Les Cardwell is an Enterprise Data Architect at Central Lincoln People's Utility District on the Oregon Coast, one of the a recipients of the ARRA Smart Grid Grants. He received his doctorate (DCS-DSS) from Colorado Technical University, and received both a MIT and BIT from American InterContinental University. Les is a subject matter expert (SME) for the DOE's Electric Subsector Cybersecurity Capability Maturity Model (ES-C2M2), is a Certified Enterprise Architect (EACOE), and an evangelist for solving the Cybersecurity challenges through an Enterprise Architecture perspective. His experience spans 30 years improving ICT efficiencies, with a passion for reducing complexity to its simplest form. Email: LesCardwell@gmail.com.

Annie Shebanow is an engineer who received her BA in computer science from the University of California at Berkeley, and her MS in business management and Ph.D. in computer science from Colorado Technical University. She is exploring how the application of big data analytic affects cyber security malicious-code detection. Shebanow is also the founder of Cloud in Exchange, a startup centering on a commodities exchange for trading cloud computing resources to improve utilization. She has founded several startup companies in the past. Shebanow volunteers as a dissertation coach for IT graduate students. As an avid advocate for female engineers, she established the Women-In-Technology Association at CTU.

Cyber Security & Information Systems Information Analysis Center

Call for Papers for Publication

The Cyber Security and Information Systems Information Analysis Center (CSIAC) - https://thecsiac.com, is one of eight Department of Defense Information Analysis Centers (IACs) sponsored by the Defense Technical Information Center (DTIC) - http://www.dtic.mil/dtic/.

CSIAC has been formed as the consolidation of three legacy IAC's – the Information Assurance Technology Assurance Center (IATAC), the Data and Analysis Center for Software (DACS), and the Modeling and Simulation Information Analysis Center (MSIAC) – along with the addition of the new technical domain of Knowledge Management and Information Sharing. CSIAC is chartered to leverage best practices and expertise from government, industry, and academia on Cyber Security and Information Technology. CSIAC's mission is to provide DoD a central point of access for Information Assurance and Cyber Security to include emerging technologies in system vulnerabilities, R&D, models, and analysis to support the development and implementation of effective defense against information warfare attacks.

CSIAC publishes the quarterly *Journal of Cyber Security and Information Systems*, focusing on scientific and technical research & development, methods and processes, policies and standards, security, reliability, quality, and lessons learned case histories. The latest issue may be viewed or downloaded at https://www.thecsiac.com/journal/welcome-csiac.

During the calendar year 2013 CSIAC will be accepting articles submitted by the professional community for consideration.

Articles in the areas of ***Information Assurance, Software Engineering, Knowledge Management, Information Sharing,*** and ***Modeling & Simulation*** may be submitted.

CSIAC will review articles and assist candidate authors in creating the final draft if the article is selected for publication. However, we cannot guarantee publication within a fixed time frame. Note that CSIAC does not pay for articles published.

To Submit an Article

Drafts may be emailed to Journal@thecsiac.com.

Preferred Formats:

- Articles must be submitted electronically
- MS-Word, or Open Office equivalent

Size Guidelines:

- Minimum of 1,500 – 2,000 words (3-4 typed pages using Times New Roman 12 pt font)
- Maximum of 12 pages, double column, including references
- Authors have latitude to adjust the size as necessary to communicate their message

Images:

- Graphics and Images are encouraged.
- Print quality, 200 or better DPI. JPG or PNG format preferred.

For the full Article Submission Policy, see page 30 of this journal.

Case Study: Applying Agile Software Methods to Systems Engineering

By Matthew R. Kennedy, PhD and David Umphress, PhD

Delivering a Software Intensive System (SIS) that is on time, within budget and with the required functionality with traditional systems processes has been problematic (Hagan 2011). This problem will only increase as the complexity of SISs within the Department of Defense (DoD) grows and more functionality within systems is relegated to software (Force 2009, Group 2009). Ultra-modern approaches— known as "agile" processes— have emerged to correlate with the rate of change encountered during software development. Agility is "the speed of operations within an organization and speed in responding to customers (reduced cycle times)" (Daniels 2006). The degree of agility when developing an IT system is the organization's ability to respond to changing requirements and technology. With quick technology refresh rates, long development cycles run the risk of placing a system in a state of obsolescence prior to initial release. The need to change without notice throughout the development lifecycle is paramount to success in the ever-changing world of technology.

Functions performed by software in DoD aircraft has increased from 8 percent for the F-4 Phantom II in 1960 to 90 percent for the F-35 in 2006 (Ferguson 2001, Schmidt 2013). With the proliferation of software within current systems, problems that were inherently software are evolving into system problems (Group 2009). The issues of both system complexity and agility is not only recognized within the United States DoD, but also have been identified in the United Kingdom's Ministry of Defense as some of the "next great systems thinking challenges" (Oxenham 2010) .

Since software has such a predominant influence on systems today, it seems natural to examine efforts within the software engineering community to control cost, schedule, and performance. The balance of this paper describes an effort to apply software agile techniques at the systems level. It describes, in the context of a case study an Agile Systems Engineering Framework, a technique developed specifically to help program managers be as agile and nimble as possible to their shifting environments.

CASE STUDY

The company used in the case study has done so under the agreement that they shall remain anonymous. To comply with this request the company will be referred to as "Juggernaut" for the purposes of this case study. The name Juggernaut has no relation to the company in the study. The product technical specifications, design documents and any detailed information that could be used to trace the case study back to the product or company will not be contained in this report.

Company Background

Juggernaut is an ISO 9001:2008 registered company with over a 100-year history and offices in multiple countries. Juggernaut produces various Information Technology (IT) solutions to customers worldwide.

Juggernaut was initially a manufacturing organization that has expanded to include manufacturing, mechanical and software departments. As their product line increased in complexity and software became a larger part of their systems, their traditional manufacturing top-down development methodology was found ineffective. Their products were becoming routinely late, over budget and did not include the planned functionality.

Small changes to the waterfall-like manufacturing process were found to be ineffective and traditional agile software processes did not provide the framework needed to incorporate manufacturing, mechanical and software components into a single delivery. Juggernaut soon realized a new development approach was required to allow for the rapid delivery of systems containing more than just software. Agile software development had been successfully adopted within Juggernaut's software department, but the need to expand those practices to include the entire systems engineering process was becoming evident.

Project Background

The project selected to utilize the Agile System Engineering Framework and Practices was comprised of hardware, firmware, software and manufacturing components. The software component was already using agile software practices. This effort was a major modification of an existing product and included the incorporation of new functionality and updates to the system's hardware, firmware, software and manufacturing elements.

Internally, Juggernaut was comprised of several departments including quality control, engineering, operations, marketing / sales and manufacturing. The engineering department consisted of multiple projects, of which the project involved in the case study was one of the ongoing projects. Each engineering project had a series of sub-elements depending on the product under development. The personnel assigned to each sub-element were not necessarily 100 percent dedicated to the one project, but were working on several projects simultaneously. Each project within the engineering department was also competing for other company resources such as manufacturing, operations, marketing / sales and / or Quality Control (QC). A system development organizational structure can be seen in Figure 1.

In addition to the internally developed hardware, firmware, software and manufacturing, portions of these components were outsourced to leverage external expertise in emerging technologies. Roughly 50 percent of the design and mechanical components were outsourced. These outsourced components needed to be accounted for and managed within the Agile Systems Engineering Framework to allow for external components to be tested at various integration points. In addition to these outsourced components, Juggernaut used manufacturing facilities that were located both in and outside of the United States (US) to assemble the final product, which increased the coordination effort required during development.

Though this product was sold directly to customers, the product under development was also intended to be reused and integrated with two other products internally developed at Juggernaut. The product needed to conform and / or be certified in several specifications, including American National Standards Institute (ANSI) and United States Military Standard (MIL-STD) specifications. The combination of internally and externally developed components coupled with the certification process made the identification and definition of the interfaces and integration points paramount to the success of the project.

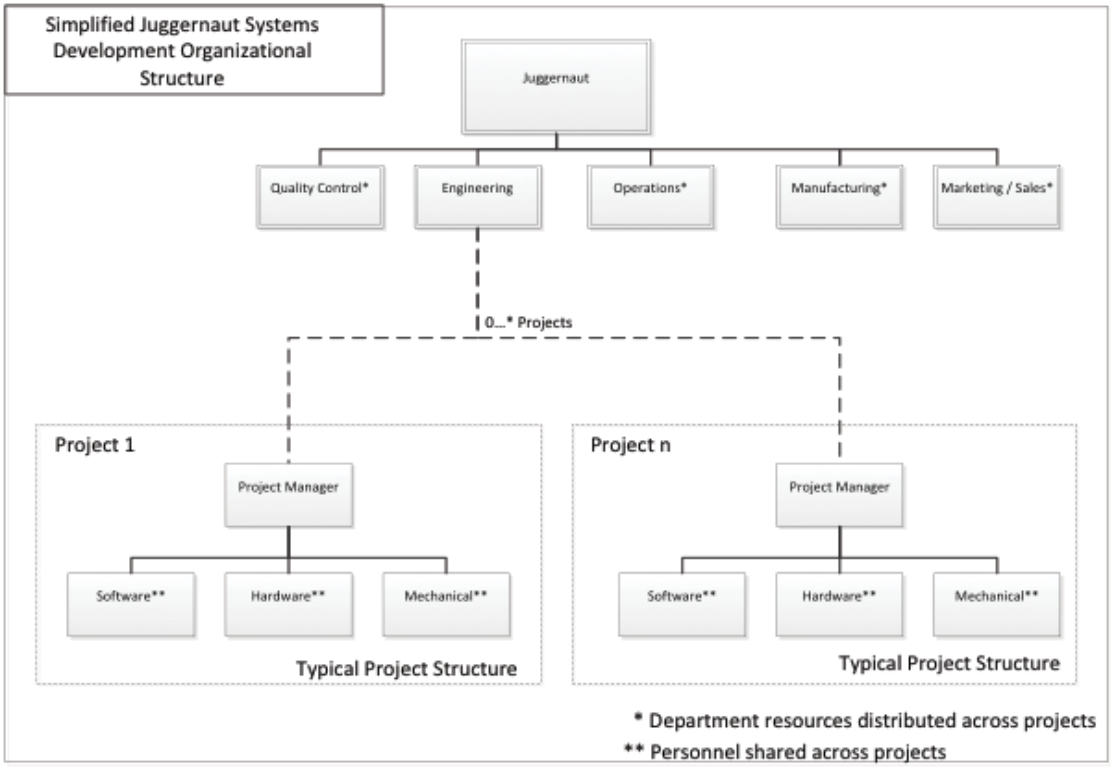


Figure 1 Juggernaut Organizational Structure

The project had a $1.3M budget for 13 months of development. The internal team, team members working directly for Juggernaut, consisted of sixteen multidisciplinary members in the following specialties: Project Management, Firmware Development, Software Development, Hardware Development and Systems Testing. The internal team worked together in the past on similar projects; however, they were utilizing a waterfall-type development methodology. This effort was the first implementation of an agile system engineering methodology employed project-wide.

Past Performance

Juggernaut was able to provide complete past performance metrics including cost, schedule and functionality delivered on twelve projects developed by Juggernaut that were of similar size and scope. Based on the twelve projects, Juggernaut was habitually behind schedule, over budget and not delivering the planned functionality. The past performance metrics were calculated by taking the average of the estimated versus actual numbers for all three data points on the twelve projects. The results can be seen in Table 1 Past Performance.

Average Cost Difference from Estimate	+2.5%
Average Schedule Difference from Estimate	+30%
Average Functionality difference from planned	-5%

Table 1 Past Performance

Identifying the Gap: The System Engineering Framework

Development of a SIS encompasses three aspects: Business, System and Software. Though there is overlap among these aspects, specific responsibilities can be attributed to each Aspect.

The Business Aspect is responsible for the acquisition of the system as a whole including contracting, funding, operational requirements and overall system delivery structure. The System Aspect is responsible for the general technical and technical management aspects of the system and serves as the interface between management and engineers. The Software Aspect is responsible for the software items contained in the SIS.

When developing a SIS, all three aspects need to work in harmony to produce a successful final product, as SISs are held captive by their slowest Aspect. General Aspect characteristics can be seen in Figure 2.

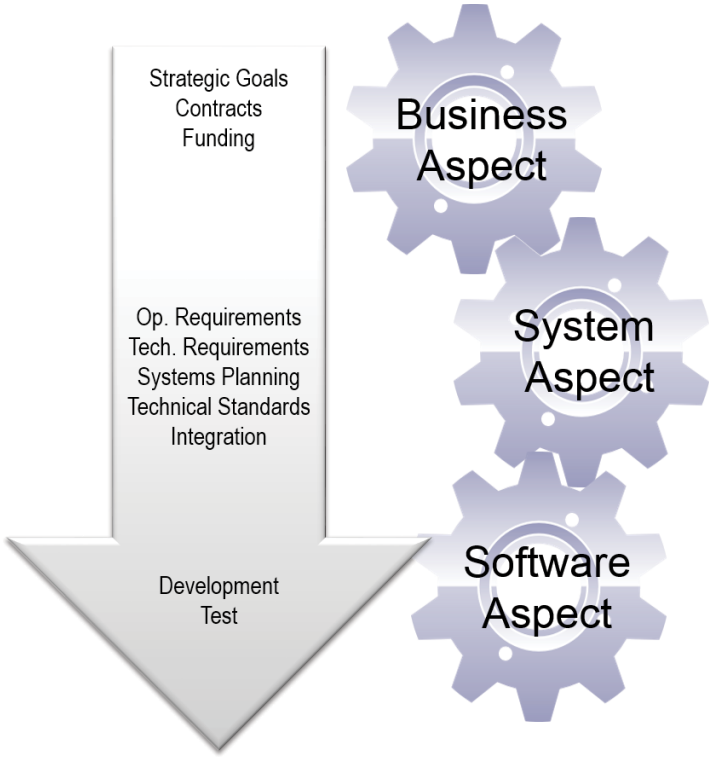


Figure 2 General Aspect Characteristics

Various agile frameworks exist in both the Business and Software Aspects. The Business Aspect has frameworks such as the Business Capabilities Lifecycle (BCL) and the Defense Science Board Agile Framework (Force 2009, Hand and Little 2012). The Software Aspect can utilize frameworks such and Scrum; however, there is a large gap between the Business and Software Aspect frameworks which does not allow for complete synchronization between all Aspects. The Business Aspect framework defines deliverables at a high level with typical delivery times of 12-18 months whereas the Software Aspect is more granular defining a single delivery in weeks. The System Aspect needs to provide a framework which allows for the management of multiple asynchronous Software Aspect deliveries to complete the desired capability defined in the Business Aspect (Figure 3). With the vast complexity of evolving SIS this framework gap does not allow for easy incorporation of system engineering best practices such as interface management, modular / open designs, configuration management or risk management.

An Agile System Engineering Framework was developed for the System Aspect to foster agile practices in the Software Aspect as well as retain the ability to rapidly respond to changes from the Business Aspect (Figure 4). This framework also

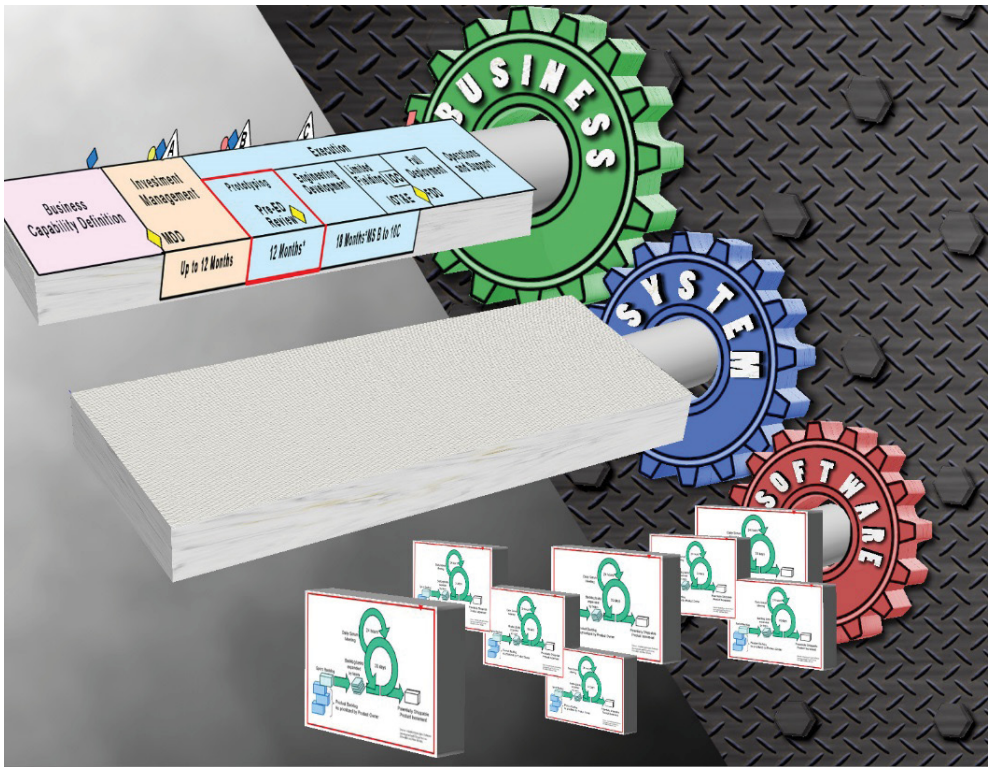


Figure 3 Missing System Aspect Framework (Graphic by Kelly Helms)

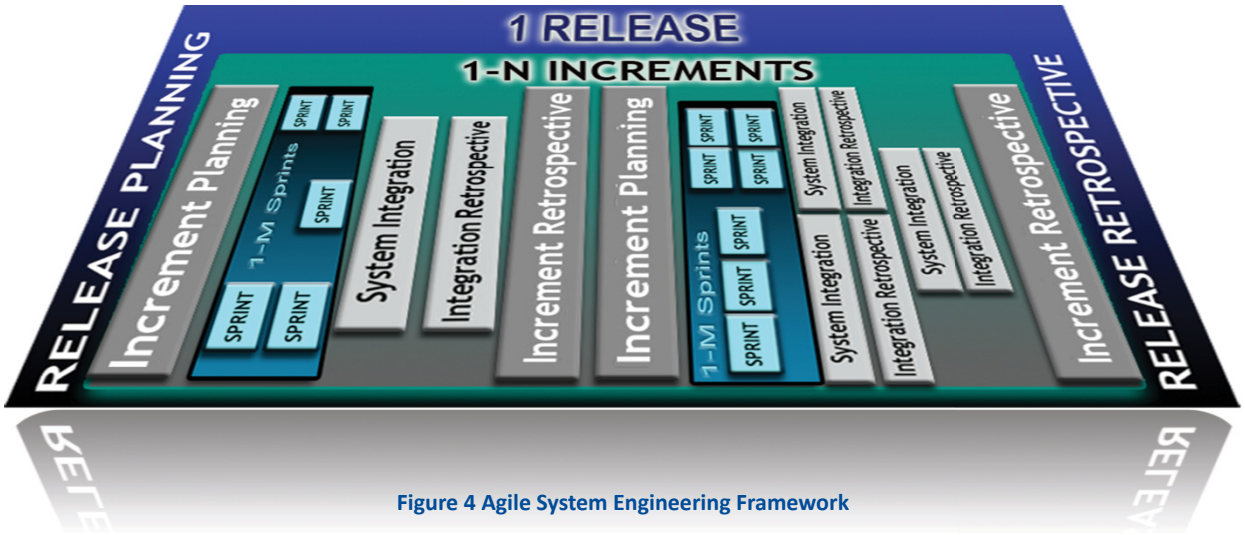


Figure 4 Agile System Engineering Framework

enables the application of systems engineering best practices to be used throughout system development.

The Agile System Engineering Framework defines three main phases: Release, Increment and Integration. Each phase is completed with a retrospective. The Agile Systems Engineering Framework is comprised of a single release divided into a series of Increments, with each Increment containing one or more Sprints and Integration blocks. A retrospective assesses each phase and provides lessons learned in order to improve the current processes the next time the phase is implemented. The Agile Systems Engineering Framework provides a good foundation for agile system engineering; however, to further incorporate system engineering best practices and agile practices further descriptions such as input / exit criteria and activates to be complete during each phase was defined to further assist in the operational implementation of the framework. These phase descriptions are defined below.

Release Phase

The product of the release is the delivered system. It starts with a Release Planning meeting(s), consists of multiple Increments and is completed with a Release Retrospective.

Input Criteria: High Level Design

Exit Criteria: Finished product to be fielded

Release Planning

Activities in this Phase

- Requirements Engineering
- Increment Time Estimation
- Identify Key System Interfaces

Exit Criteria: Prioritized Release Backlog

Increment Phase:

The Increment Phase receives the prioritized Release backlog from Release Planning Phase. The output of an Increment is an item that is placed under configuration management. Each Increment consists of one or more Sprints and Integration phases.

Input Criteria: Prioritized Release backlog

Exit Criteria: Finished "Configuration Item"

Increment Planning

Input Criteria: Prioritized Release Backlog

Activities in this Phase

- Decompose the system into functional items
- Identify high risk items
- Identify Key System Integration Points
- Identify / further define Key System Interfaces
- Specify temporal dependencies among Sprints; i.e., determine which Sprints can be conducted concurrently and which must be conducted sequentially
- Select what can be done at each Increment based on the prioritized release backlog
- Identify the personnel / resources / skill set that should be involved in the Increment
- For each Sprint
  - Identify 'customer'
  - Specify the definition of 'Done'
  - Specify / Identify expected outputs / specifications for each Sprint

Exit Criteria

- Prioritized Sprint backlog(s)
- Incremental program plan identifying Sprints and Integration points

Sprint

A Sprint consists of a time-boxed window for producing a potentially shippable product to be integrated into the system in the parent Increment or Integration Phase. The Sprint block is where development of any kind occurs and is handled as a black box within the Agile System Engineering Process. Here, a form of "black box trust" occurs allowing each Sprint to develop the specified product freely provided the product is completed using the minimum required specifications and interfaces provided in the Input Criteria. The Sprint development risk is managed by a combination of the input specification / interfaces and the development time-boxed window. In general, the shorter the development time, the less the investment. A product could include software and / or hardware but may also be items like a Commercial-Off-The-Shelf (COTS) product evaluation assessment. Multiple Sprints can be underway concurrently.

Input Criteria

- Sprint Backlog
- Specifications / Interfaces
- Customer Identification
- Definition of "Done"

Activities in this Phase: Item Development

Exit Criteria: Completed / user-accepted product(s)

System Integration Phase

The integration phase combines various elements of the overall SIS. These elements could be a combination of hardware and / or software produced by the Sprints and / or the incorporation of Government-Off-The-Shelf (GOTS) / Commercial-Off-The-Shelf (COTS) products required by the SIS. This is the phase where high risk pieces can be incrementally constructed to assess the feasibility of the combination of components.

Input Criteria: Completed items and specifications / interfaces for the systems to be integrated from previous Sprints, Increments and / or GOTS / COTS.

Activities in this Phase

- System Integration
- Specifications / interfaces validation / review / refinement

Exit Criteria: User accepted, integrated system or subsystem

Juggernaut was provided with the Agile System Engineering Framework and Descriptions to facilitate the gap between the Business and Software Aspect (Figure 5). In addition,

Incremental Development	Small Teams
Iterative Development	Time-Boxing
Short Time-lines	Lean Initiatives
Retrospectives (Lessons learned)	Prototyping
Empowered/ Self-organizing/ Managing teams	Continuous User Involvement
Prioritized Product Backlog (Requirements)	Co-located Teams

Table 2 Agile Systems Engineering Practices (Kennedy and Ward 2012)

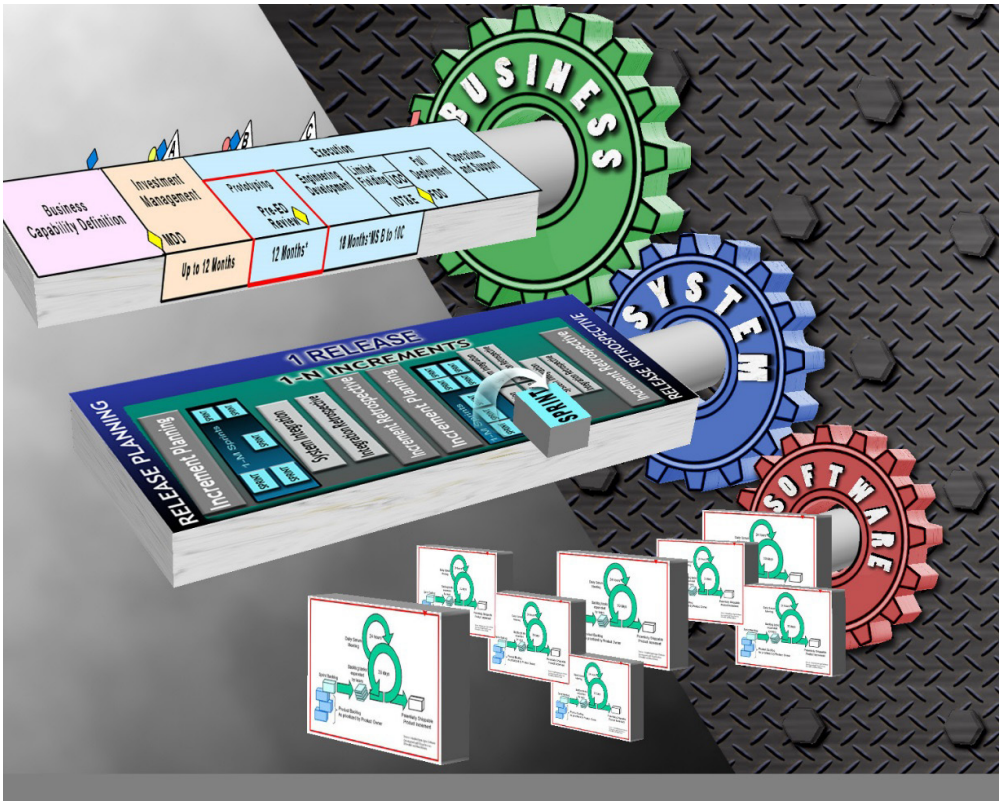


Figure 5 Complete Framework Interaction (Graphic by Kelly Helms)

Juggernaut was provided a list of agile practices found effective in both the Business and Software Aspects (Table 2).

The Framework and Practices were delivered to Juggernaut in two face-to-face training sessions. During these meetings the case study primary researcher was also provided an overview of the product under development. In addition to the training meetings, quarterly teleconferences and virtual meetings were held to provide progress and feedback on the implementation of the agile systems engineering process. Information was also collected and shared through email correspondence which totaled over 75 email exchanges throughout the development process.

Upon acceptance of the Framework and Practices, Juggernaut reengineered its project plan to conform to the new framework. Based on analysis from the Release Planning Phase, three Increments were planned, each containing Sprints that were designed to coincide with key integration points in which a combination of internally and externally developed hardware, firmware, enclosures and / or software needed to be integrated and tested by the Quality Control (QC) group.

At these key integration points, the hardware and software were required to have certain functionality and the QC group had to have the necessary outside resources, equipment and personnel to run specific integration tests. As defined in the

process, each integration point had a set of predetermined input and exit criteria used to measure whether a successful integration was accomplished.

Reorganizing the project into the new structure allowed for the identification of critical system components and interfaces. High risk items could be identified and completed early and integration risks could be mitigated by performing incremental integration throughout development. The new structure also allowed for more accurate project tracking since each sprint constituted a completed item as defined by the predetermined "definition of done" versus attempting to track items using an estimated percent complete methodology. No item should be given credit until it is 100 percent complete.

Framework Implementation Example

During the Increment planning phase Juggernaut designated their Sprints as hardware, mechanical, software and / or firmware sprints. Each Integration Phase required input from a predetermined number of sprints and / or external dependencies (designs, hardware components, COTS products, etc.). The framework implementation example from the case study requires input from three internally developed Sprints, though other Integration Phases may require both internal and external inputs to achieve the specified objectives. In this example details have been removed such as the actual specification requirements and additional details required to adequately test the exit criteria for instance required response times and detailed accuracy ranges. Names of the customer point of contact are also not included. The example is intended to provide the overall organization Juggernaut used to structure their systems development using the Agile Systems Engineering Framework. A graphics depiction can be seen in Figure 6.

Integration 1: Wired Functionality Objectives

- Decision on the Solar Panel / Internal Battery concept
- Metrology meets specified tolerance ranges
- Hardware Interface meets current profile requirements
- Initial wire testing is complete with list of defects
- Validation of Test Point access (Test Engineering)

Input Criteria:

Sprint 1 – Hardware

Input Criteria:

- Hardware Specification Document

Exit Criteria:

- Support multiple energy sources
- Discrete solution for the register interface

Sprint 2 – Mechanical

Input Criteria:

- Design Specification Document

Exit Criteria:

- Rolled Sealed Assemblies
- Housings According to Specification

Sprint 3 – Firmware

Input Criteria:

- Firmware Specification Document

Exit Criteria:

- Wire Communication Standard Conformance
- Liquid Crystal Display (LCD) Functionality
- LCD Activation Level
- Metrology Functionality

Exit Criteria

- Current Consumption / Profile
- Wire Interface HW Testing
- Primary Power Supply Capabilities
- LCD Testing
- E&M Field Testing
- Electrostatic Discharge Testing
- LCD Activation Level (multi-source)Wire Response
- Clock Detection Accuracy
- ASIC Read
- Operational Compliance Testing
- Environmental Testing

Project Results

For case study tracking purposes, the project was divided into three critical milestones: Milestone 1 marked the successful completion of internal QC; Milestone 2 was the release to a

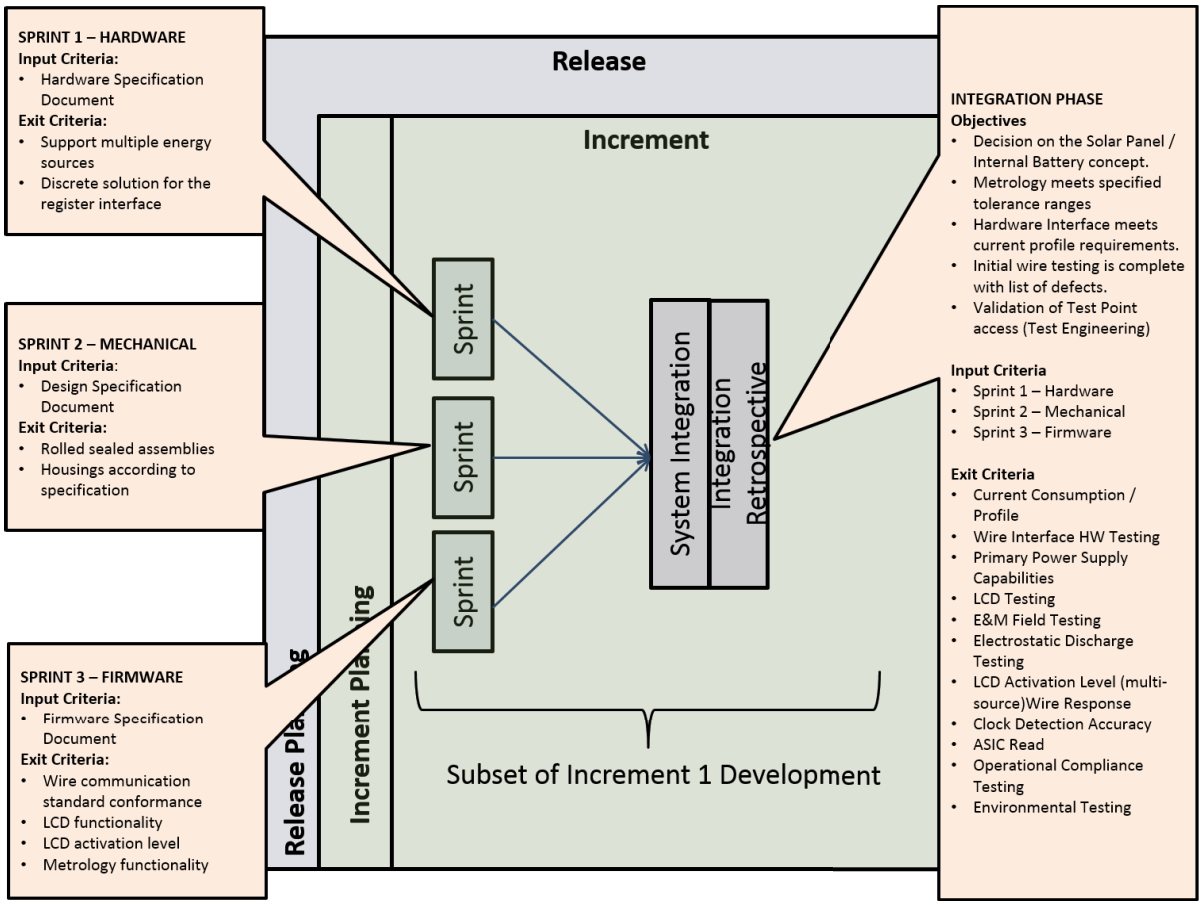


Figure 6 Example Framework Implementation

limited user base and external testing; and Milestone 3 was mass production and customer sales. The metrics for this case study focus on the first milestone.

Completion of Milestone 1 was the focal point of this case study as it included design, development and internal QC of the product. Juggernaut produced seventy units during this phase to put through internal QC. After the successful completion of the first milestone, the product specifications were sent to several production plants in various countries for manufacturing resulting in 500 Low Rate Initial Production (LRIP) units. After receiving the results of the user feedback and QC tests, Juggernaut would perform an assessment to determine if the units were ready for production (thus concluding Milestone 2), then move into Milestone 3, mass production and customer sales.

The completion of Milestone 1 was scheduled for 27.5 weeks and the actual completion took 29 weeks. This was a 5.5 percent increase in duration from the initial estimate. When compared to the past performance of Juggernaut, there was a 24.5 percent improvement in predicting their schedule using the Agile Systems Engineering Framework and Practices.

Since Juggernaut had been developing similar systems for years, their cost estimation was typically accurate prior to using the Agile Systems Engineering Framework and Practices. Throughout the project, the teams were assigned a "cost goal" based on the overall cost estimate. These cost goals were then used by the teams to make tradeoffs throughout the project. The teams were able to meet their cost goal by balancing cost factors such as scope, material costs or labor. On this project, the largest contributor to the under run in budget was that one of Juggernaut's vendors agreed to decrease their profit margin resulting in a decrease in overall project cost. Based on Juggernaut's past performance metrics, Juggernaut typically ran 2.5 percent over budget. At the completion of Milestone 1, Juggernaut was 5 percent under budget marking a 7.5 percent difference in cost estimation. Because the cost was due to a vendor renegotiation, the cost fluctuation was not attributed to the Framework or Practices utilized during product development. Without the vendor renegotiation the cost savings was estimated to be less than 1 percent.

Typically Juggernaut experienced a decrease in 5 percent of the planned functionality in order to better meet their cost and schedule goals. At the completion of Milestone 1, they delivered 100 percent of the planned functionality showing an overall improvement of 5 percent.

	Past Performance	*New Model*	*Result*
Cost Difference from Estimate[1]	N/A	N/A	N/A
Schedule Difference from Estimate	+30%	+5.5%	24.5% Improvement
Functionality difference from planned	-5%	0%	5% Improvement

Table 3 Case Results Data

The benefits of the Framework and Practices were seen outside of the engineering division and stretched into marketing. The increase in predictability of delivery dates allowed the marketing division to better plan for the marketing aspect of the product.

Juggernaut successfully completed Milestone 2 and produced 4000 units. Approval for mass production was achieved and is currently underway.

Summary

The Agile Systems Engineering Framework and Practices do not remove typical project management issues encountered during systems development, but they enable early identification and resolution of issues. Juggernaut encountered many of the same issues faced by projects regardless of the Framework and / or Practices used during systems development. Issues include:

1. Scheduling priorities – Other project took priority in manufacturing, testing, or development;
2. Staffing issues;
3. Fluctuation in material costs;
4. Manufacturing lines shut down;
5. Delays in receiving ordered parts;
6. Retesting of subsystems during development due to unsuccessful QC.

Though Juggernaut faced these issues during development, by structuring their project into the specified Framework and using the Agile Practices, they were able to identify, restructure and adapt to these issues with minimal impact to the overall project. Juggernaut experienced some unanticipated events that caused minor schedule slips to occur. One of the initial "phase reviews" was completed ahead of schedule, which was noted by management as the "first time in company history".

The incorporation of the Agile Systems Engineering Framework and Practices showed an improvement in estimating the systems cost, schedule and functionality in addition to reinforcing systems engineering best practices such as interface management, configuration management, risk management and overall technical management. Implementing agility is a different puzzle for each system. Identify your puzzle and SOLVE IT!

References

Daniels, J. J. (2006). Review of Acquisition for Transformation, Modernization, and Recapitalization, U.S. Army War College. **Master of Strategic Studies:** 28.

Ferguson, J. (2001). "Crouching dragon, hidden software: software in DoD weapon systems." Software, IEEE **18**(4): 105-107.

Force, D. S. B. T. (2009). Department of Defense Policies and Procedures for the Acquisition of Information Technology. Washington, D.C. 20301-3140**:** 109.

Group, S. (2009, 04/23/2009). "Standish Newsroom - CHAOS 2009." New Standish Group report shows more project failing and less successful projects. Retrieved 01/01/2011, 2011, from http://www1.standishgroup.com/newsroom/chaos_2009.php.

Hagan, G. (2011). Glossary of Defense Acquisition Acronyms and Terms. Fort Belvoir, Virginia 22060-5565, Defense Acquisition University Press**:** 346.

Hand, W. L. and G. C. Little (2012). Guide To The Business Capability Lifecycle For Department Of Defense ACAT III Programs, LMI Research Institute**:** 76.

Kennedy, M. R. and D. Ward (2012). "Inserting Agility in System Development." Defense Acquisition Research Journal **19**(3).

Oxenham, D. (2010). Agile approaches to meet complex system of system engineering challenges: A defence perspective. System of Systems Engineering (SoSE), 2010 5th International Conference on.

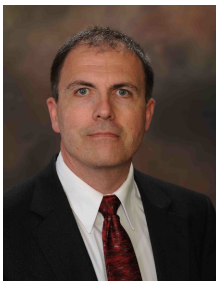
Schmidt, D. C. (2013). "The Growing Importance of Sustaining Software for the DoD." Retrieved 02/16/2013, 2013, from http://blog.sei.cmu.edu/post.cfm/the-growing-importance-of-sustaining-software-for-the-dod.

(Footnotes)

1 The cost variance was due to a vendor renegotiation and was not attributed to the Framework or Practices.

About the Author(s)

Matthew R. Kennedy is a Professor of Software Engineering at Defense Acquisition University (DAU). Previously he was the Associate Director of Engineering at the National Cancer Institute's Center for Biomedical Informatics and Information Technology (CBIIT). He served in the U.S. Air Force as a network intelligence analyst and has more than 13 years of experience in Information Technology. He has a Bachelor's and Master's degree in Computer Science and a PhD in Computer Science and Software Engineering.



David A. Umphress, Ph.D., is an associate professor of computer science and software engineering at Auburn University, where he specializes in software development processes. He has worked over the past 30 years in various software and system engineering capacities in military, industry, and academia settings. Umphress is an Institute of Electrical and Electronics Engineers (IEEE) certified software development professional.

Software Protection Against Side Channel Analysis Through a Hardware Level Power Difference Eliminating Mask

By John R. Bochert, Michael R. Grimaila, and Yong Kim

Side Channel Analysis (SCA) is a method by which an adversary can gather information about cryptographic keys by examining the physical environment surrounding the microprocessor while it is performing cryptographic operations. In this article, we present our research which is focused upon devising methods to increase the difficulty of conducting SCA successfully on a microprocessor running Advanced Encryption Standard (AES) encryption. We make use of the open-source, soft-core Java Optimized Processor (JOP) implemented on a Xilinx Virtex 5 ML506 Field Programmable Gate Array (FPGA) evaluation board to evaluate the effectiveness of SCA countermeasures in attacks against the cryptographic algorithm. The experimental results show that implementing a power normalizing mask can increase the security of a device by requiring an adversary to collect up to 87% more data to successfully attack AES.

I Introduction

Security and cryptography in electronics have played an integral part in society for several decades. Starting with securing military communication channels and in the civilian sector with Automated Teller Machines (ATMs), the need for security has been on the rise for decades. Secure crypto-processors in particular (microprocessors that process cryptographic algorithms) have become the backbone of modern security solutions. One can find crypto-processors in smart cards, cable and satellite TV set top boxes, lottery ticket vending machines, and mobile-phone systems. As adversarial techniques and skills have evolved to compromise crypto-processors, so have the means used by manufacturers to protect or prevent system tampering, reproduction, disabling, and reverse-engineering [3].

There are basically four different classes of attack by which an adversary can attack a crypto-processor: Semi-Invasive Attacks, Invasive Attacks, Remote Attacks, Local Noninvasive Attacks [3]. In this section, we briefly review the attributes of each of the attack classes. Semi-Invasive Attacks do not require damaging the coating of the semiconductor surface, known as the passivation layer, because it uses lasers to ionize atoms within the transistors and change its state. This method is difficult to employ in practice due to the variability inherent when attempting to ionize specific transistors making information extraction unreliable. Invasive Attacks involve actual damage to the device and monitoring of the device interior. Although this can be useful to gain information, it also destroys the device which is unacceptable when an adversary only has a limited number of devices, or only a single device, to analyze. Remote Attacks interface with a device in normal operation over a communication channel such as exploiting a buffer overflow exploit in a networked device. Remote attacks have their place, but they deal solely with programming vulnerabilities and not hardware vulnerabilities. Local Noninvasive Attacks involve gaining information about the device through close observation of the device in operation, watching Electromagnetic (EM) radiation emissions, current consumption, and other environmental effects surrounding the device. Local Noninvasive Attacks were chosen to be the focus of this research because of the magnitude of the risk they pose. They allow attackers to circumvent cryptographic algorithms just by having physical access to the device. Side Channel Analysis (SCA) attacks, characterized as Local Non-Invasive attacks, are the method by which an adversary can

cleverly deduce information about a cryptographic system by watching the interaction of a circuit with its surrounding environment. The three main branches of SCA are timing, power-analysis, and EM attacks. In all types, the basic idea is to determine a cryptographic device's secret key by measuring its execution time, power consumption, and/or electromagnetic field [16].

In this paper, we present the findings of our initial research focused upon improving the security of cryptographic processors. The goal of our research is to propose new methods to protect cryptographic information by making dynamic changes to the underlying architecture of a microprocessor. To measure the effectiveness of different protection methods, we implemented the Advanced Encryption Standard (AES) cryptographic algorithm in a soft-core Java Optimized Processor (JOP) contained within an FPGA and measured the time required to expose the underlying cryptographic key using standard SCA methods.

II. Related Work

A. Background

The basic premise of SCA attacks stem from the reality that the switching activity of Complementary Metal-Oxide Semiconductor (CMOS) circuits leak information. When a CMOS circuit charges to logic level '1' or discharges to logic level '0', a change in the electric potential creates a change in the electric field (or current) which is measurable outside the chip. Generally the quantization of the energy for a given value is derived from either the Hamming Weight (HW) or the Hamming Distance (HD). In the case of the HW, the value of a given data is the summation of the bits that are in a 'non zero' state. For example, the HW of 0x50 (0b01010000) and the HW of 0x03 (0b00000011) are both two. The HW of 0xFF (0b11111111) is eight. In contrast, the HD is a measure of the change of a value, measuring the number of bits that change from the previous state to the current state. For example the hamming distance between 0x50 and 0x03 is four, while the hamming distance between 0x50 and 0xFF is six. The Hamming Weight can also be thought of as the Hamming Distance between the given value and zero (0x00). Commonly, the model used to describe the information leakage off a chip is given by $C(t)^{(a,b)} = \lambda HW\left(a \oplus b\right) + \beta_t$, where $(a \oplus b)$ is the "exclusive or" (XOR) of a and b, HW is the Hamming Weight function, λ is the power consumption used by the circuit when inverting the bit, and β_t is the noise [4].

After monitoring the execution time, power consumption and/or the electric field from a microprocessor, the three main branches of SCA attacks used to find secret key information are: Simple Power Analysis (SPA), Differential Power Analysis (DPA), and Second Order Differential Power Analysis (SODPA) [14]. A SPA attack involves directly observing a system's power consumption and can be done with only one trace. DPA is significantly more powerful than SPA, but is more complicated and requires the collection of many more traces. DPA looks at the changes in the trace values over time to narrow down using statistical hypothesis testing. DPA is normally done by looking at the difference of means or using Correlation Power Analysis (CPA). Lastly SODPA is a method often used to overcome many time variable masking countermeasures. It involves looking at the values of traces at several points in time for a trace so that all of the mask will be accounted for when various correlation methods are used.

Defenses against these SCA attacks fit into two high-level categories: algorithmic countermeasures (changes made to the algorithm of encryption) and circuit-level countermeasures (changes made to the actual hardware). Countermeasures can be further classified based on the method by which they try to decouple the power consumption with the data being processed, these are: masking countermeasures (trying to make data appear as a different value) and elimination countermeasures [14] (trying to remove any correlation of the data being processed and the power signatures being measured).

B. Masking Techniques

Many masking techniques at the circuit level introduce random power consumptions which are akin to noise. Examples include Random Switching Logic (RSL) [15], masking-AND [18], and Dynamic Voltage and Frequency Switching (DVFS) [19]. The RSL countermeasure adds in random logic paths, masking-AND masks every output with random inputs, and DVFS randomly modulates voltage and switching frequency to introduce randomness into power traces. All of these circuit level masking techniques, however, are still susceptible to glitches. Glitches are the transitions at the output of a gate that occur before the gate switches to the correct output. Because glitches add to the power signature, they are susceptible to leak information, especially when they leak key information before the correct mask is applied [9] [2]. RSL uses random input and enable control signals to randomize the power signature and is thus able to avoid the information leakage posed by glitching, but the enable signals need to be carefully timed to ensure it functions properly [2].

Masking at the algorithmic level has the key notion of minimizing the correlation between intermediate values and

the secret key [5]. One simple method to accomplish this is to introduce noise into the power consumption measurements. This method can be overcome by the collection of more samples. In theory, if the variance of the noise is great, then the necessary sample size might be large and infeasible. However, this method is still surmountable by increasing the number of samples used in the analysis [7].

Another option for masking power traces at the algorithmic level is the introduction of Random Process Interrupts (RPI) during the cryptographic algorithm. This approach can be done by interleaving random dummy commands or "No Operation" (NOOP) commands randomly throughout the code thus masking the actual cryptographic algorithm execution sequence. To attack a circuit using RPIs, the correlation spikes can be reconstructed by integrating the signal over the number of consecutive clock cycles equal to the greatest variance in the clock cycles [6]. This method to overcome RPIs is called the "sliding window attack." For this attack, several traces are integrated together and then compared against other integrated traces for the power spikes [8].

A more common method of masking however is to split a value Z into d shares $M_1 \Diamond ... \Diamond M_d$ such that $M_1 \Diamond ... \Diamond M_d = Z$ and where $\Diamond$ is a function like the XOR or modular addition [12]. A masking operation is said to be $(d-1)^{th}$-order depending on the number of shares d. When a $(d-1)^{th}$ order masking is used, a d^{th}-order DPA can be performed by combining the leakage signals at time intervals $L(t_1), \ldots, L(t_d)$ resulting from the manipulation of the d shares that make up the value Z. This method of masking generally can be circumvented through the use of higher-order differential power analysis. By combing the leakage signals at time intervals $L(t_1), \ldots, L(t_d)$ that are the resulting leakages from the manipulation of the d shares that make up the value Z, the differential power spike for correct key guesses can be reproduced [12][10][11].

C. Elimination Techniques

Elimination is another method that can be used to confound power variation. The key notion of elimination (hiding) is to remove power variation information from the attacker. Where masking seeks to decouple the power variation from the data being processed, elimination seeks to eliminate it. Four ways that elimination is used to protect circuitry are [7]:

- Using constant execution path code
- Choosing operations that leak less information in their power consumption
- Balancing hamming weights and state transitions
- By physically shielding the device

With the goal of elimination techniques to be no variation in power due to the key, no information is leaked through side channels. One example of this is Dynamic and Differential Logic (DDL), where elimination of power differences is done through ensuring that one of the outputs is charged for any input, be it the output or the complimented output, and it ensures that one output transition occurs in every clock cycle. More specifically, DDL logic is split into a precharge state, where all outputs are at zero, and then an evaluation phase, where at least one output or its compliment goes high [17]. Sense Amplifier Based Logic is an implementation of DDL that uses dynamic-CMOS logic. Using this method of DDL, requires the circuit designer to deal with the effects of cascading circuits and can introduce signal integrity issues that degrade the signal making it inefficient [17].

Wave Dynamic and Differential Logic (WDDL) is another type of DDL. WDDL uses a static CMOS implementation of AND and OR gates. Each gate in the WDDL has both the gate with the inputs and a complimentary gate with the inverse of the inputs. By introducing complementary structures, the information that is leaked via the side channel is reduced.

D. Advanced Encryption Standard (AES)

Advanced Encryption Standard (AES), the cryptography algorithm used in this research, is a symmetric key crypto-algorithm. A symmetric cryptographic algorithm uses the same key to both encrypt and decrypt data. The AES algorithm is made up of eleven rounds. With the exception of the first and last round, each round is made up of the four functions: SubBytes, ShiftRows, MixColumns, and AddRoundKey. Of particular note for this research are SubBytes and AddRoundKey. SubBytes uses a simple substitution algorithm and takes the current hex values and substitutes the values with known quantities in a look-up table called a "substitution box" (also known as the Sbox). AddRoundKey is the function where the key, modified slightly for each round, is XORed with the current text state [1].

Of those four functions, only AddRoundkey directly manipulates the data based on key, which makes AddRoundkey the target for key extraction in SCA. The first call to AddRoundKey uses the original key, and each subsequent call to AddRoundKey uses a different version of the key. Following the AddRoundKey phase is SubBytes which uses a simple substitution algorithm where by the current state of the plain text is used to find the corresponding substitution value in the Sbox. The best place to attack the AES algorithm is between the AddRoundKey phase of the

previous round and the SubBytes of the next round. This location is highly vulnerable because given the plaintext, an attacker knows exactly what the state of the plaintext is going into AddRoundKey, but he does not know what the state will be after returning from AddRoundKey as they do not yet know the key. The attacker does, however, know the simple look-up table used in SubBytes and if he can correlate power signatures to approximate values, it's possible with enough traces to use a statistical algorithm to derive what the key is. The specific location of AES attack is shown in Figure 1.

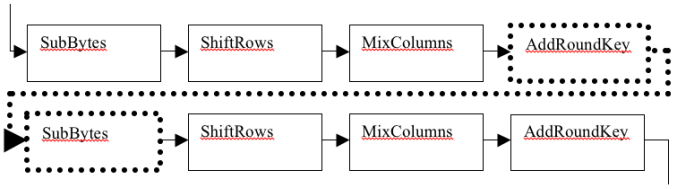


Fig. 1. Two rounds of AES with location of attack noted with dotted lines

III. Methodology

In this research, we made use of a Java Optimized Processor (JOP) which is a soft-core CPU written in VHDL. One of the key benefits of the JOP is that it can be implemented on most commercially available FPGAs with no modifications. Internally, the architecture of the JOP is a 4-stage, pipelined CPU with separate Bytecode Fetch, Fetch, Decode, and Execute/Stack stages. These stages control the operation of the memory controller, the cache, the ALU, and the I/O interface contained within the JOP. The JOP was designed primarily to provide an efficient Java interpreter for embedded systems. The JOP was chosen for this research because it is open source (e.g., free) and uses a modern CPU pipelined architecture. The use of a soft-core JOP enabled us to easily implement different hardware protection schemes by modifying the underlying JOP architecture. In our research, we implemented the AES algorithm in the Java programming language and executed the Java code on three different variations of the JOP. A secret key was randomly generated and the same secret key was used for all of the experiments. In order to implement the hardware countermeasures, changes were made to the underlying VHDL architecture of the JOP [13]. These countermeasures were then evaluated as using side channel analysis.

A. Side Channel Analysis (SCA)

Side Channel Analysis (SCA) is used to gain information about the secret key by measuring EM emissions while the AES algorithm is executing. The emissions are considered the "signal" of interest for an attacker and arise from the movement of data between the processor and memory as the AES algorithm executes. Specifically, as operations are performed within the JOP, data needed for the operations is stored and retrieved from a local cache to improve processor performance. Interestingly, the cache in the JOP is implemented in the form of a stack. As data is moved between the registers and the cache, information about the key is leaked in the form of EM emissions which are correlated with the key. Worse, when these cached values are written back to main system memory, the EM emissions often have greater magnitude due to the larger capacitances present in external data buses which interface the processor and main system memory. Thus, for circuits running AES cryptography, protection methods are centered on reducing emissions resulting from data transfers between the processor, cache, and RAM write-back operations. With this in mind, our hypothesis is that through the addition of memory units using both masking and elimination techniques, the correlation between the processed data and the EM emissions can be significantly reduced.

B. A Protection Method for a Java Optimized Processor

In the protected version of the JOP, when values are saved, they are first split into two values. First, all of the odd bits of the original value, *D*, are saved in the first part of the mask, *D0*, as all the odd bits with each even bits containing the inverse of the odd bits. Thus, the first bit of *D* (b0) would be stored in the first bit of *D0*, and the second bit of *D0* would be the inverse of the first bit (b1'). Then the third bit of *D* (b2) is stored in the third bit of *D0*, and the fourth bit of *D0* would be the inverse of the third bit (b2'). The second part of the mask, *D1*, would contain all of the even bits of the original value, *D*, and all of the odd bits of *D1* would contain all of the inverse values of the even bits. See Figure 2 below.

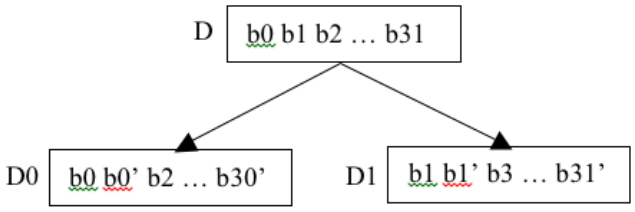


Fig. 2. The separation of a 32 bit integer into two values of equal hamming weight

This method of masking splits the values into two masked values, and for the 32 bit system of the JOP each of the resulting masked values will always have a Hamming Weight (HW) of 16 regardless of what the original value was. The HW is the number of bits in the number in the "on" position. This method to cause every value to have the same HW helps eliminate power usage differences for different values. It is

implemented when values are stored to the JOP cache and implemented when values are stored in the RAM.

C. Testing Setup

The testing setup for this experiment used an ML506 Virtex 5 FPGA running AES algorithm on a soft-core JOP. Electromagnetic emissions, hereafter called "traces," were collected using a RISCure EM probe connected to a Lecroy Wavemaster 8zi oscilloscope and analyzed using RISCure's "Inspector" software package (see http://www.riscure.com/). The physical test setup is shown in Figure 3.

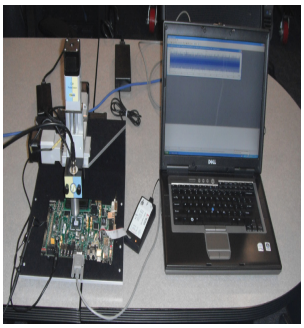


Fig. 3. Testing setup showing EM probe centered over Virtex 5 with shield removed

In order to find the best location to center the EM probe before collecting data, the chip surface was divided into a 10x10 grid. EM measurements were collected at each of the 100 locations to determine the location with the best signal to noise ratio. Figure 4 shows an example color graph depicting the magnitude of the power levels recorded over the surface of the chip. In this graph, light green is the area of greatest signal to noise ratio.

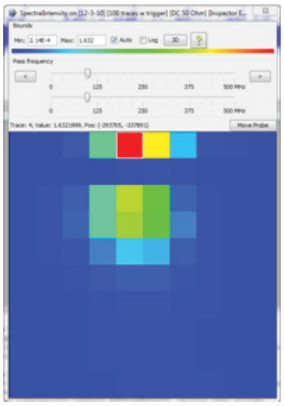


Fig. 4. Physical surface of the chip showing locations of high EM radiation

Once the optimal location for the EM probe was selected, 1000 traces were collected during the AES encryption of random plaintexts, one trace per encryption. Because the VHDL code for each of the variations of the JOP was different (e.g., baseline JOP architecture with no countermeasures, JOP with masked cache architecture, and JOP with masked RAM architecture), the process above was repeated to determine the optimal probe location in order to maximize the signal to noise ratio for each of the three architectures. From these 1000 traces, a 1st-order DPA analysis was performed considering only the portion of the trace that occurred in SubBytes during the first round of AES. The "Inspector" software tool was used to perform the DPA analysis. This software requires the user to provide the plaintext and identify the relevant portions of the collected trace to analyze. The software generates a testable statistical model in which the collected data is used to test a set of hypothesis until the statistically most probably key emerges. In every case of the 1000 trace set, the correct key was found. Subsequently, the process was repeated using a smaller number of traces until the correct key was not found. This procedure yielded the minimum number of needed traces to deduce the correct key. This whole process was repeated thirty times to obtain the *average* minimum number of traces required to deduce the correct key. These thirty data points, derived from the 30,000 collected traces, represented the number of needed to arrive at the correct key using DPA for a given JOP architecture.

Figure 5 below shows an example trace, recorded from the EM probe during one encryption of AES in the baseline JOP with no protections. The periodic nature of the 10 rounds of AES can be easily seen in the collected trace.

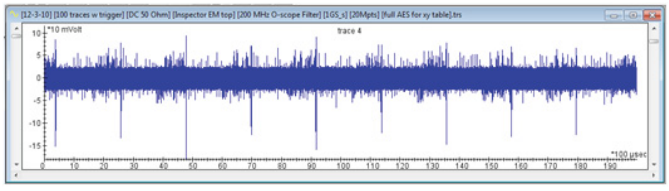


Fig. 5. A trace of AES showing 10 rounds in the unprotected baseline JOP

Figure 6 shows the areas of interest when the SubBytes of the first AES round occurs. Specifically, Figure 6(a) highlights the trace when the cache is being accessed and Figure 6(b) highlights the traces when RAM write backs occur. These areas represent the targets of opportunity for reducing the signal available to an attacker during DPA.

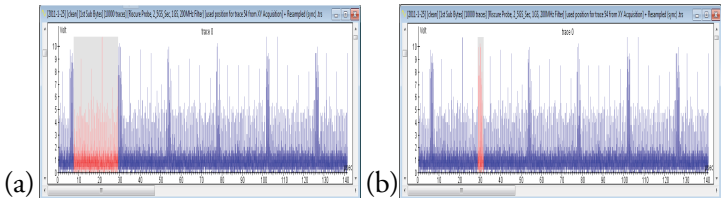


Fig. 6. (a) The execution portion of a SubBytes substitution and (b) the RAM write back portion

IV. Results

In summary, the DPA presented in this research used traces collected during the SubBytes phase of the first AES round for three different versions of the Java Optimized Processor (JOP): 1) unprotected baseline JOP, 2) a JOP with masked cache, and 3) a JOP with masked RAM. Figure 7 shows the traces collected from each of the three JOPs during the SubBytes phase of the first AES round. Figure 7(a) shows the traces collected from the baseline unprotected JOP; Figure 7(b) shows the trace collected from the JOP with the masked cache; and Figure 7(c) shows the trace collected from the JOP with the masked RAM.

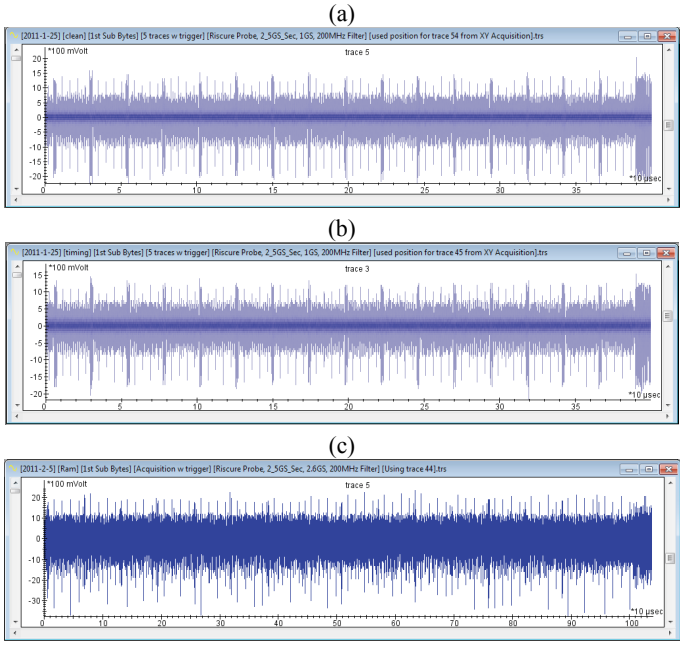


Fig. 3. AES SubBytes for unprotected JOP (a), masked cache (b), masked RAM (c)

When attacking the JOP without any countermeasures, the execution portion of the trace (with heavy cache usage) required an average of 308.3 traces to extract the correct key, while the RAM write-back portion of the trace required an average of 154.8 traces. The execution portion of the trace required more traces than the RAM write-back portion because less power is used by the JOP to interface with the on-chip cache than the off-chip RAM, thus providing less leakage and weaker signals from the JOP.

When using the masked cache countermeasure, a t-test with a 95% confidence showed the increase in security to be between -95 traces to 14 traces during the execution stage. This means that with a 95% confidence, there is no statistical increase in security. This was found to be due to the fact that the JOP contains many registers that pass the values and communication between registers was not protected by the masked cache, so information was still leaked.

When using the masked RAM countermeasure, a 95% confidence t-test of the data found that the average increase in the needed traces to find the correct key increased from 43 traces to 137 traces. This means that with a 95% confidence, the masked RAM had a substantial improvement in the security for the RAM write-back portion of the trace. This gives us an increase in the number of needed traces to derive the correct key to be between 31% and 87% with a 95% confidence. As expected, there is significantly greater leakage of information during the RAM write back than during the transfer of data from the registers to the cache, requiring about half the needed traces as compared to considering information leaked by the cache. This clearly indicates that efforts at reducing the leakage of the RAM write back module will yield the best return on investment when protecting a processor.

However, it is important to note that masking schemes incur costs in terms of die area and processor speed. The costs of the cache protection was negligible, having less than 1% total increase in area of the CPU and having no impact on processor performance. In contrast, the RAM masking scheme increased the execution time by 2.5x what it was previously, and required twice the RAM. Determining if these costs are acceptable depend upon the specific application context.

IV. Conclusions

In this research using the AES encryption in a JOP, we found that implementing a masked cache did not yield a statistical increase in security while implementing a masked RAM did have a measureable difference in security. The masked RAM with a 95% confidence interval showed that the increase in security (as shown by the number of traces required to find the correct key) was between 31% to 87%. This increase in security as compared to the same method applied to the cache with no increase is due largely to the fact that the RAM uses greater power than the registers and cache and leaks more

information. Thus protecting this portion of the JOP has a greater effect than the protected cache did. However, masking RAM incurs a significant penalty in performance and requires additional RAM blocks to implement.

The lack of security increase for the protected cache was that the on-chip registers were not protected and leaked as much information as the cache did, thus the increase in security due to the protected cache was negligible. To correct this problem, the underlying data structure of the JOP would need to be changed. Currently the JOP employs a Von Neumann architecture where both the instructions and the data are both saved in the same memory. If the JOP structure was instead changed to a Harvard Architecture, where the instructions and data are saved in two different locations, it would be possible to split the data values and save them split in the double RAM, and keep them split as they move through the JOP all the way to the execute phase of the CPU when they would be "reassembled" as they're being used for calculations. This is not currently feasible in the JOP because when a value is read from the RAM, instructions and data are indistinguishable and obfuscating instructions would require significant changes to the decoding stage. Changing the underlying architecture could reasonably increase the protection of the JOP several orders of magnitude, making the JOP 100 or 1000 times more secure.

Disclaimer

The views expressed in this paper are those of the authors and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the U.S. Government.

References

FIPS 197, "Federal Information Processing Standards Publication 197 : Announcing the Advanced Encryption Standard (AES)," National Institute of Standards and Technology (NIST), November 26, 2001.

Alam, M., Golsh, S., Mohan, M.J., Mukhopadhyay, D., Chowdhury D.R., and Gupta I.S., "Effect of Glitches against Masked AES S-box Implementation and Countermeasure," *IET Information Security*, 1 Oct, 2008.

Anderson, R., Bond, M., Clulow, J., and Skorobogatov, S., "Cryptographic Processors - A Survey," University of Cambridge Computer Laboratory Technical Report UCAM-CL-TR-641, ISSN 1476-2986, 94(2), February 2006.

Aumonier, S., "Generalized Correlation Power Analysis," Oberthur Card Systems SA, 2007.

Chari, S., Jutla, C.S., Rao, J.R., and Rohatgi P., "Towards sound approaches to counteract power-analysis attacks," Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '99) Springer-Verlag, 398–412, 1999.

Clavier, C., Coron, J.S., and Dabbous, N., "Differential power analysis in the presence of hardware countermeasures," Lectures Notes in Computer Science, 1965:252–263, 2000.

Kocher, P., Jaffee, J., and Jun, B, "Differential Power Analysis," Lecture Notes in Computer Science, CRYPTO 1999; 1666:388–397, 1999.

Lu, Y., O'Neill, M. P., and McCanny J.V., "FPGA Implementation and Analysis of Random Delay Insertion Countermeasure against DPA," ICECE Technology, FPT 2008 International Conference, Dec 2008.

Mangard, S., Popp, T., and Gammel B.M., "Side Channel Leakage of Masked CMOS Gates," CT-RSA 2005, The Cryptographers' Track at the RSA Conference, 3376:351–365, 2005.

Messerges, T.S., Dabbish, E.A. and Sloan, R.H., "Examining Smart- Card Security under the Threat of Power Analysis Attacks," IEEE Transactions on Computers, 51(5), May 2002.

Nohl, K., Evans, D., Starbug, and Plotz H., "Reverse-Engineering a Cryptographic RFID Tag," USENIX Security Symposium, Jul 2008.

Prouff, E., Matthiew, R., and Bevan R., "Statistical Analysis of Second Order Differential Power Analysis," Transactions on Computers, 58(6), June 2009.

Schoeberl, M., "Java Optimized Processor," http://www.jopdesign.com/.

Sundaresan, V., Srividhya, R., and Vermuri R., "Defense against Side-Channel Power Analysis Attacks on Microelectronics Systems," Aerospace and Electronics Conference, 2008. NAECON 2008. IEEE National, 2008.

Suzuki, D., Saeki, M., and Ichikawa, T., "Random Switching logic: A New Countermeasure against DPA and Second-Order DPA at the logic level," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, E90-A(1):160–168, 2007.

Popp, T., Oswald, E., and Mangard, S., "Power Analysis Attacks and Countermeasures," IEEE Design and Test of Computers, 535–543, 2007.

Tiri, K., Akmal, M., and Verbauwhede, I., "A Dynamic and Differential CMOS Logic with Signal Independent Power Consumption to Withstand Differential Power Analysis on Smart Cards," Proc. of European Solid State Circuits Conference (ISSCIRC 2002), 403–406, 2002.

Trichina, E. "Combinational logic design for AES subbyte transformation on masked data," Cryptology e-print archive: Report 2003/236, IACR, Nov 2003.

Yang, S., Wolf, W., Vijaykrishnan, N., Serpanos, D.N., and Xie, Y., "Power Attack Resistant Cryptosystem Design: A Dynamic Voltage and Frequency Switching Approach," Proc. of Design Automation and Test in Europe Conference (DATA 2005), 351–365, 2005.

About the Author(s)

Capt John R. Bochert, USAF, received his BS in Electrical Engineering from the Air Force Academy (AFA) in 2007 and his MS in Computer Engineering from the Air Force Institute in Technology (AFIT) in 2011 where worked on cryptography and VLSI systems. Capt Bochert is a member of the IEEE and his research interests include computer programming, parallel programming, network systems, graph theory, and finding evidence for God in science.

Dr. Michael R. Grimaila (BS 1993, MS 1995, PhD 1999, Texas A&M University) is an Associate Professor of Systems Engineering and Management and member of the Center for Cyberspace Research (CCR) at the Air Force Institute of Technology (AFIT), Wright-Patterson AFB, Ohio USA. He is a Certified Information Security Manager (CISM), a Certified Information Systems Security Professional (CISSP), a member of the ACM, a Senior Member of the IEEE, and a Fellow of the ISSA. Dr. Grimaila's research interests include quantum information and cryptography, mission assurance, network management and security, and systems engineering.

Dr. Yong C. Kim (BSCE, University of Washington, 1995; MSECE, University of Wisconsin, 1997; PhD, University of Wisconsin, 2002) is a Senior Electronics Engineer at the Air Force Research Laboratory, Wright-Patterson AFB, Ohio USA. His areas of interest are hardware assurance, computer architecture, VLSI design, and fault-tolerant computing.

we like your feedback

At the CSIAC we are always pleased to hear from our journal readers. We are very interested in your suggestions, compliments, complaints, or questions. Please visit our website http://journal.thecsiac.com, and fill out the survey form. If you provide us with your contact information, we will be able to reach you to answer any questions.

Article Submission Policy

The CSIAC Journal is a quarterly journal focusing on scientific and technical research & development, methods and processes, policies and standards, security, reliability, quality, and lessons learned case histories. CSIAC accepts articles submitted by the professional community for consideration. CSIAC will review articles and assist candidate authors in creating the final draft if the article is selected for publication. However, we cannot guarantee publication within a fixed time frame.

Note that CSIAC does not pay for articles published.

AUTHOR BIOS AND CONTACT INFORMATION

When you submit your article to CSIAC, you also need to submit a brief bio, which is printed at the end of your article. Additionally, CSIAC requests that you provide contact information (email and/or phone and/or web address), which is also published with your article so that readers may follow up with you. You also need to send CSIAC your preferred mailing address for receipt of the Journal in printed format. All authors receive 5 complementary copies of the Journal issue in which their article appears and are automatically registered to receive future issues of Journal. Up to 20 additional copies may be requested by the author at no cost.

COPYRIGHT:

Submittal of an original and previously unpublished article constitutes a transfer of ownership for First Publication Rights for a period of ninety days following publication. After this ninety day period full copyright ownership returns to the author. CSIAC always grants permission to reprint or distribute the article once published, as long as attribution is provided for CSIAC as the publisher and the Journal issue in which the article appeared is cited. The primary reason for CSIAC holding the copyright is to insure that the same article is not published simultaneously in other trade journals. The Journal enjoys a reputation of outstanding quality and value. We distribute the Journal to more than 30,000 registered CSIAC patrons free of charge and we publish it on our website where thousands of viewers read the articles each week.

FOR INVITED AUTHORS:

CSIAC typically allocates the author one month to prepare an initial draft. Then, upon receipt of an initial draft, CSIAC reviews the article and works with the author to create a final draft; we allow 2 to 3 weeks for this process. CSIAC expects to have a final draft of the article ready for publication no later than 2 months after the author accepts our initial invitation.

For some issues CSIAC has a Guest Editor (because of their expertise) who conducts most of the communication with other authors. If you have been invited by a Guest Editor, you should

PREFERRED FORMATS:

- Articles must be submitted electronically.
- MS-Word, or Open Office equivalent (something that can be edited by CSIAC)

SIZE GUIDELINES:

- Minimum of 1,500 – 2,000 words (3-4 typed pages using Times New Roman 12 pt font) Maximum of 12 pages
- Authors have latitude to adjust the size as necessary to communicate their message

IMAGES:

- Graphics and Images are encouraged.
- Print quality, 200 or better DPI. JPG or PNG format preferred

Note: Please embed the graphic images into your article to clarify where they should go but send the graphics as separate files when you submit the final draft of the article. This makes it easier should the graphics need to be changed or resized.

CONTACT INFORMATION:

CSIAC
100 Seymour Road Suite C102
Utica, NY 13502
Phone: (800) 214-7921
Fax: 315-351-4209

John Dingman, Managing Editor
Phone: (315) 351-4222
Email: jdingman@quanterion.com

Tom McGibbon, CSIAC Director
Phone: (315) 351-4203
Email: tmcgibbon@quanterion.com

CSIAC is a DoD sponsored Information Analysis Center (IAC), administratively managed by the Defense Technical Information Center (DTIC), technically managed by the Air Force Research Laboratory (AFRL) in Rome, NY and operated by Quanterion Solutions Incorporated, Utica, NY.

ABOUT THE JOURNAL OF CYBER SECURITY AND INFORMATION SYSTEMS

Distribution Statement:
Unclassified and Unlimited

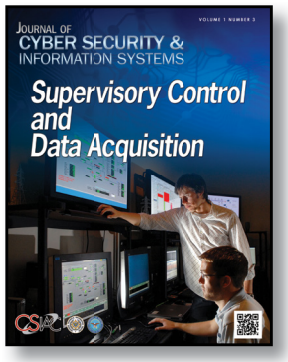
CSIAC
100 Seymour Road
Utica, NY 13502-1348
Phone: 800-214-7921 • **Fax:** 315-732-3261
E-mail: info@thecsiac.com
URL: http://www.thecsiac.com/

ABOUT THIS PUBLICATION

The Journal of Cyber Security and Information Systems is published quarterly by the Cyber Security and Information Systems Information Analysis Center (CSIAC). The CSIAC is a DoD sponsored Information Analysis Center (IAC), administratively managed by the Defense Technical Information Center (DTIC). The CSIAC is technically managed by Air Force Research Laboratory in Rome, NY and operated by Quanterion Solutions Incorporated in Utica, NY.

COVER DESIGN

Shelley Howard
Graphic Designer
Quanterion Solutions, CSIAC



From the Cover: LOGIIC (Linking the Oil and Gas Industry to Improve Cyber Security) team members Bryan Richardson and Weston Henry demonstrate the project's monitoring solution hosted at Sandia's Center for Control Systems Security. (Photo by Randy Montoya)

ARTICLE REPRODUCTION

Images and information presented in these articles may be reproduced as long as the following message is noted:

"This article was originally published in the Journal of Cyber Security and Information Systems Vol. 1, No 3 May 2013."

In addition to this print message, we ask that you notify CSIAC regarding any document that references any article appearing in the *CSIAC Journal.*

Requests for copies of the referenced journal may be submitted to the following address:

Cyber Security and Information Systems
100 Seymour Road
Utica, NY 13502-1348

Phone: 800-214-7921
Fax: 315-732-3261
E-mail: info@thecsiac.com

An archive of past newsletters is available at **https://journal.thecsiac.com.**

Cyber Security and Information Systems Information Analysis Center
100 Seymour Road
Suite C-102
Utica, NY 13502

Return Service Requested

Journal of Cyber Security and Information Systems – May 2013
Supervisory Control and Data Acquisition

— IN THIS ISSUE —