# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*



**PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.**

| 1. REPORT DATE *(DD-MM-YYYY)* | 2. REPORT TYPE | 3. DATES COVERED *(From - To)* |
|---|---|---|
| 01/15/2013 | Final Performance Report | March 2009 - November 2012 |

**4. TITLE AND SUBTITLE**
Proof-Carrying Survivability

**5a. CONTRACT NUMBER**

**5b. GRANT NUMBER**
FA9550-09-1-0215

**5c. PROGRAM ELEMENT NUMBER**

**6. AUTHOR(S)**
Zuo, Yanjun

**5d. PROJECT NUMBER**

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
University of North Dakota
293 Centennial Drive Stop 8363
Grand Forks, ND 58202

**8. PERFORMING ORGANIZATION REPORT NUMBER**

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
Air Force Office of Scientific Research
875 N. Randolph Street
Arlington, VA 22203
Dr. Robert Herklotz/RSL

**10. SPONSOR/MONITOR'S ACRONYM(S)**

**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**
AFRL-OSR-VA-TR-2013-0066

**12. DISTRIBUTION/AVAILABILITY STATEMENT**
Distribution A: Approved for Public Release

**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT**

The objective of this project was to develop a logic-based framework for proof-carrying survivability: a user publishes his/her survivability requirement policy; a system provider constructs a proof to show that the system satisfies the user's requirements; and the user verifies if the proof is valid. We have developed techniques that enable users to specify measureable and certifiable survivability requirements and represent their survivability policy. From the perspective of system acquisition and engineering, developing a survivability requirement is the important first step in survivability specification, compliance formulation, and proof verification. An important part of the project is a logic model for survivability requirement representation, compliance proof construction, and user verification. We have developed various techniques for proof evidence search, proof derivation, and proof code checking. Moreover, the logic model facilitates constrained reasoning, i.e., possibilistic uncertainty and survivability requirement constraints are effectively linked to logical reasoning. Our work makes it possible to express fuzzy pattern matching and

**15. SUBJECT TERMS**
Survivability, requirement specification, logical framework, proof generation and verification, constrained reasoning, RFID

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | c. THIS PAGE | | | Yanjun Zuo |
| U | U | U | U | | 19b. TELEPHONE NUMBER *(Include area code)* (701) 777-2517 |

Reset

**Final Performance Report**

Submitted to

The Air Force Office of Scientific Research

By

The University of North Dakota

**Proposal Title**: Proof-Carrying Survivability

**Grant Number**: FA9550-09-1-0215

**Principal Investigator**: Dr. Yanjun Zuo

**Program Manager**: Dr. Robert L. Herklotz

## 1. MAJOR ACCOMPLISHMENTS

The major accomplishments of the project include:

- Attack modeling and survivability quantification of critical systems
- A Bayesian decision model to help users better understand the hidden states of a critical system in order to determine its survivability status based on prior knowledge and current available evidence
- A simulation model using Colored Petri Net to measure the degree of survivability given varying compromise and recovery rates in different attacking scenarios
- Methodology for reasoning about users' survivability requirements
- A flexible, balanced, and threshold-based approach for survivability requirement specification
- A proof-carrying survivability logic including a logic language and its formal semantics
- Algorithms and techniques to automatically generate a proof and verify the proof
- A systematic approach to facilitate the system provider in collecting proof evidence and identifying the most efficient proof collection schedule to execute
- A constraint-annotated logic to incorporate constraints to logical reasoning, which reflects the inherent, non-trivial relationships among the survivability properties required for a system
- Fuzzy pattern matching and constrained reasoning in proof construction and verification
- Simulations and prototyping of the proof-carrying survivability models
- Techniques and methodology for survivability and security of distributed enterprise RFID systems

## 2. EXECUTIVE SUMMARY

Given the increasing complexity of cyber attacks and the ultimate importance of system survivability, we have developed a comprehensive framework for proof-carrying survivability: a system user publishes his/her survivability requirements, a system provider compiles and submits

a proof-carrying code, and finally the user applies a simple and fast program to verify the proof. If the proof is validated, all the requirements are guaranteed to be satisfied. By shifting the responsibilities of the survivability proof from a user to a system provider, the user only needs a lightweight checker to verify that a system possesses a set of safety and security properties for system survivability. Any system that does not meet the user's requirements will be detected before the system is deployed.

Our work included two main thrusts. First, we developed techniques for the user to quantify system survivability. Attack modeling and survivability quantification allow the user to identify the critical system properties necessary for a system to survive malicious attacks and system failures. The survivability requirements are specified based on the qualitative analysis of system properties and modeling of attacks. We studied the inherent survivability properties of critical systems in general and distributed enterprise Radio Frequency Identification (RFID) systems specifically.

In the second thrust, we developed the logic constructs for proof-carrying survivability. An application-specific logic was designed with sound formal properties. Specifically, the logic framework facilitates constrained reasoning, i.e., possibilistic uncertainty and survivability requirement constraints are effectively linked to logical reasoning. The framework makes it possible to express fuzzy pattern matching and arbitrary user-defined constraints in formal proofs.

The above two major thrusts of research are discussed in more detail in the following sections. Furthermore, as RFID has emerged as an important technique for many high security and high integrity settings, we used RFID as a running example to illustrate our techniques. As a result of this project, some technical breakthroughs on RFID survivability and security are also reported in this document.

### 2.1 Survivability Quantification and Requirement Specification

From the perspective of system acquisition and engineering, survivability requirement is the important first step in survivability specification, compliance formulation, and proof verification. Rigorous survivability analysis needs both qualitative and quantitative approaches to produce a holistic view of the survivability features that a system must have in order to withstand malicious attacks and system failures and provide mission-critical services. We have proposed techniques and approaches for system survivability requirement analysis and specification as discussed next.

We started with attack modeling and system survivability quantification. An abstract survivability experiment is specified as a foundation in order to measure the degree of survivability of a critical system under varying attacks. Stripping details of system security protocols and attacks to their necessity and applying formal analysis allow us to study the survivability strength and weakness of those protocols under attack. Based on this foundation, a series of malicious scenarios is modeled using stochastic process algebras, and the different effects of those attacks on the ability of the system to provide critical services are studied. Due to built-in fault tolerance and recovery, damages to some components of a system do not necessarily mean that the entire system is not survivable. Therefore, the survivability level of a system is measured given two opposite effects: the compromise of individual components and the recovery of damaged components. Stochastic process algebras introduce timing and

probability qualifications to pure process algebras. The timed and probabilistic process algebras are well suitable for modeling the concurrent, dynamic interactions between the adversary and a system with uncertainty (such as the random time for a component to be compromised). Formulas were developed to measure the survivability level of the system given the occurrence of the system at each state. Our model has an underlying Markov chain, and we model the stochastic process algebra using the Performance Evaluation Process Algebra (PEPA) tool. One of the major objectives of attack modeling and survivability quantification is to study the significant impact of different attacks on system survivability and provide solid data for users in their survivability requirement specification.

To better understand the survivability status of a system in different attacking stages, we also analyzed the system states given the adversaries' compromise ability and the system's recovery ability. Our study shows that the system under attack evolves through the following states: *clear state* -> *safe state* -> *survivable* or *unsurvivable state*. We also identified the critical points for each state transition. To verify the analysis, we developed a simulation model using Colored Petri Net to (1) simulate the survivability status of a system given a set of attack-recovery scenarios and (2) to measure the degree of survivability given varying compromise and recovery rates. The model shows how sensitive the degree of survivability is to both a system's recovery ability and an adversary's attacking capability. The analysis and the simulation model provide guidelines for system administrators to identify the system's survivability under a given attack-recovery process as well as the technical mechanisms to improve the system's survivability.

Given the increasing complication of malicious attacks and the increasing complexity of the systems, it is often difficult to determine whether a critical system has been compromised or is in a faulty state. It is also challenging to determine whether such a critical system should be allowed to continuously function given its uncertain state. Therefore, it is important to maintain the survivability of the system and make timely decisions on system repair, if necessary, in order for the systems to support critical services. A Bayesian decision model has been developed to help system administrators better understand the hidden states of a critical system in order to determine its survivability status based on prior knowledge and current available evidence. The model is based on our perception that the survivability of a system is dependent on several factors in such a way that probabilistic relationships exist between these factors and the system's survivability status. We represent such probabilistic relationships using a Bayesian network. Our model is used to determine whether it is adequate for the system to continue supporting critical services or whether it needs to be repaired to avoid further losses. Therefore, the model helps system administrators to reduce the magnitude of possible service interruptions due to malicious attacks or system failures.

A holistic approach has been proposed to integrate the above techniques into a comprehensive, implementable framework for survivability quantification and requirement specification. There are two major phases of activities in the framework: *phase one* – system survivability property identification; and *phase two* – survivability requirement quantification and specification. The first phase provides a qualitative view of survivability requirements for a critical system in terms of the survivability properties that the system must have in order to be considered survivable. In order to systematically identify the important survivability properties of a system, it is necessary to formally characterize the system, pinpoint its critical components and major access points, and

specify the survivability threats to the system and the critical components. The second phase is to quantitatively measure how the system features of the survivability requirement specification determine the survivability level of the system. This phase provides quantification that indicates whether the proposed survivability requirements will result in a level of survivability of the system that meets users' criteria. Two issues are addressed in this phase: (1) the cause-effect relationship between a survivability requirement specification for the system and the rate at which the critical components could be compromised; and (2) the survivability level of the system given the compromise rate of the critical components and the system's recovery capability. This quantification helps system administrators answer various what-if questions. For instance, if we change some survivability requirements for the critical system, can we still ensure that the system satisfies users' survivability criteria? Can we make some aspects of the system's survivability features stronger, and in the meantime relax some other requirements while the system can still maintain a satisfied level of survivability? What will the survivability requirements be given a different adversary capability, after the threat model shows additional evidence about the possibility of the attacks? The answers to those questions can make the system design and survivability specification measurable and flexible.

Survivability quantification and system state analysis provide a foundation for users to specify their requirements for systems to be evaluated. We have developed a decision model that enables users to specify measureable and certifiable survivability requirements and represent their survivability policy. Key variables defined include survivability characteristics, primitives, and mechanisms. A user's requirements for system survivability can be specified in two ways. For an essential survivability characteristic, the user must mandate that at least a certain number of key survivability primitives of that characteristic be satisfied. For other survivability characteristics, the user may just want to express their most tolerable level of concerns regarding certain unfavorable features. As long as those unfavorable features do not go beyond a certain intolerable level, the system is considered to satisfy the user's requirements from the perspective of those survivability characteristics.

Based on the above idea, a flexible, balanced, and threshold-based approach was developed for a user to express his/her survivability requirements with different levels of details and for the system developers to show different features of their system in order to prove that the system satisfies the user's requirements. The approach allows the user to balance both survivability requirements and some (unavoidable) concerns regarding certain system properties which may conflict with tight security control. In engineering terms, more control means additional system layers. More layers mean slower performance and higher implementation costs. Therefore, the approach is flexible so that the user can require that a system meet the criteria for some mandatory, critical system survivability features; in the meantime, the system will not have any unfavorable properties that cannot be tolerated. The proposed approach investigates such issues as how a survivability requirement policy is developed, how a survivability policy is represented and interpreted, and what the domain-dependent variables and rules are that a system provider can use in compiling a compliance proof. Since survivability compliance is critical in defining a user's survivability requirements, the proposed approach supports "provability", i.e., the system provider can compile and submit a proof to show that their system satisfies the user's survivability requirements.

## 2.2 Logic-based Proof-Carrying Survivability

Based on the threshold-based survivability requirement model, we have developed a logical framework for survivability requirement representation, compliance proof construction, and user verification. In the proposed proof-carrying survivability (PCS) framework, the user accepts the system (or system component) only if the system provider can prove that the system satisfies the survivability requirements specified as the user's policy. To show their system's compliance to the policy, the system provider needs to compile and submit a proof. Proof generation relies on the certifications generated by trusted evaluators. Therefore, the system provider first needs to collect evidence from the trusted evaluators who can confirm that the system has the required survivability features. Then, it applies the appropriate strategies and tactics to construct a proof. Finally, the user verifies whether the proof is valid. If so, the system can be considered satisfactory and acceptable. In the PCS framework, the user's survivability requirements are represented in a formal logic with application specific operators and inference rules. Proof code is generated by a prover program using the logic. Proof verification is efficient since there is no decision procedure and it is just a mechanical checking process. The system user only needs to apply a trusted checker program to verify that the proof indeed proves the published survivability requirement policy. When the scope of systems becomes large and their complexities continue to grow, PCS provides a valuable tool for users to automatically verify important system survivability properties.

The PCS logic language includes types and formulas. The basic types include character, string, integer, list, and basic connectors (e.g., arithmetic and logic connectors). The principles, the system under evaluation, the set of survivability primitives, and the survivability characteristics are all represented as strings. Formulas form the language that a survivability requirement policy and the proof terms are written in. PCS is designed using a semantic approach – each operator is defined in the underlying higher-order logic and each inference rule is proved a lemma. Since every logic term of PCS is reduced to terms of the underlying logic, the soundness of the object logic depends on the soundness of the underlying logic. Logic inference rules of PCS are used to compile and verify a survivability compliance proof. The rule set includes survivability logic specific rules and general higher-order logic rules (e.g., *and_i*, *and_e, forall_i*).

To facilitate the system provider in compiling a valid proof, a tactical prover has been developed to generate a derivation of a goal statement by applying a set of inference rules of PCS. A tactic reduces a goal to a set of sub-goals to prove. We have shown that the prover satisfied two important characteristics of a logic program – completeness and termination. To reduce the size of the code which must be trusted, the trusted computing base of the PCS framework only includes the core PCS logic and the proof checker program. The prover program is not necessarily trusted since any mis-compiled proof (by a possibly corrupted prover) cannot be accepted by the trustworthy proof checker.

Since proof evidence search is a critical step for the system provider to compile a proof, we have developed a systematic approach to facilitate the system provider in collecting evidence (e.g., system property certificates from trusted evaluators) and identify the most efficient (optimal) proof collection schedule to execute. Proof collection scheduling determines which system properties to be assessed and in which order those properties will be evaluated by the authorized evaluators. As we mentioned earlier, compiling a valid proof in the PCS framework relies on the

certifications generated by authorized evaluators indicating that the system under evaluation possesses certain properties and hence satisfies the user's requirements. Therefore, the system provider must collect evidence from the trusted evaluators to confirm that the system has the required survivability features. Algorithms and techniques have been proposed for the system provider to collect proof evidence by exploring different proof choices and options. The proof search process (evidence collection) proceeds between the system provider and the trusted evaluators interactively, and the search space is updated dynamically when new information is available. Any partial results of the previous proof attempt will be incorporated into the following proof searches to automatically reassess the next optimal proof schedule. Formal analysis shows that our approach terminates for any valid input (e.g., a user survivability requirement policy) and always generates the optimal solution if one exists, or reports a failure if otherwise.

A new formalism, i.e., a constraint-annotated logic (CAL), has been developed, where arbitrary user requirements and constraints for system survivability features can be represented and reasoned in a logical framework. CAL is to empower PCS in order to represent and reason effectively about the inherent, non-trivial relationships among the properties of a system that are required to satisfy. There could be various constraints on the properties that a user wants and needs to define. Technically, some properties may take others as pre-requisites or some properties must be satisfied at the same time. From the requirement engineering perspective, a user may want to restrict some system features due to reasons such as compliance to technical standards, limitation of budgets or organizational policies. Appropriate constraints on those and other types of restrictions are both practically necessary and theoretically important. In our research, a formal design of constraint domain has been specified so that user-defined constraints can be enforced by prohibiting logical inferences that would violate these constraints. CAL allows a user to capture a wide variety of constraints of practical interest, including atomicity, dependency, and mutually exclusive inclusion.

Technically, the CAL model defines a constraint structure to specify constraints, to create proof obligations when some constraint rules are applicable to a survivability property to be proved, and to generate resources to solve those proof obligations. To make the integration of constraint checking and logical reasoning more efficient, we have designed a mechanism for constraint specification and enforcement using the logic inference rules in such a way that creating and solving proof obligations are linked to the logical reasoning process as directed by the inference rules of the CAL logic. We have proved that CAL is consistent and constraint-complete. Furthermore, admissibility of cut can be established for CAL. Experiments and analysis show that the CAL model is a powerful formalism in reasoning hybrid domains between users' constrained requirements and system survivability properties.

To empower PCS with the ability to reason on incomplete, vague or ill-known information, we have developed techniques to unify possibilistic uncertainty with logical reasoning. The result is a comprehensive logical framework, which, together with the CAL mechanism, significantly empowers the PCS logic to represent and reason with uncertain information under a set of arbitrary, user-defined constraints. We call it constrained, possibilistic PCS framework, denoted as $\mathcal{P}$-PCS. Reasoning on incomplete or vague knowledge as supported by $\mathcal{P}$-PCS can represent the case when an evaluator is only able to assess a system's survivability property with a partial

certainty (belief degree) instead of a full level of guarantee. $\mathcal{P}$-PCS also allows an evaluator to sign a statement with a partial truth value about a system property (gradual many-valued formulas). Therefore, the truth of a formula in $\mathcal{P}$-PCS is no longer binary. Instead of being completely true or false, such a formula is labeled with a truth necessity, an element of a totally-ordered, bounded scale, representing a lower bound on the degree of necessity for the formula to be true.

We have developed the necessary logic and constraint constructs to represent and support partial belief-constrained reasoning. With variable weights and fuzzy constants defined based on formula necessities, $\mathcal{P}$-PCS makes it possible to express fuzzy pattern matching in formal logic proof. Thus, it allows representing and reasoning with uncertain information given a set of arbitrary, user-defined constraints on system survivability requirements. The interplay between constraint checking/verification and logical reasoning is through a set of logic inference rules. In summary, the main contributions of the $\mathcal{P}$-PCS framework are: (1) a formal approach to represent possibilistic uncertainty on many-valued formulas annotated with principals' belief degrees; (2) a systematic mechanism to incorporate vague and incomplete knowledge to logical reasoning; (3) a full language specified for survivability requirement constraints and constraint solving in a logic proof context; and (4) a specification of the formal properties of the $\mathcal{P}$-PCS logic and related proof search methodology.

### 2.3 Radio Frequency Identification (RFID) Security and Survivability

RFID has been used in various high security and high integrity settings including military supply chain, homeland security, transportation, and utility management (to name a few). We have achieved some technical breakthroughs on RFID survivability and security as a result of this project. They are briefly reported in this section.

RFID offers opportunities for real-time item identification and inventory tracking. For applications using resource-restricted RFID tags and mobile hand-held readers, however, various risks could threaten the abilities of an RFID system to provide essential services to users. High mobility of the RFID system components and the open nature make an RFID system vulnerable to various attacks. We studied and identified RFID survivability issues and future research in four areas: (1) developing efficient, game-changing software and hardware solutions suitable for RFID tags; (2) improving physical security which could significantly enhance physical protections of RFID components; (3) building agile, adaptive, robust, and resilient RFID systems; and (4) developing RFID fault tolerance, recovery, and intrusion detection techniques.

The development of RFID has led to many applications in the military domain. Unlike in other fields, non-survivability in military could lead to delay of troop deployments, loss of life, and even loss of a war. Since the military often operates in a more challenging environment, survivability of the essential systems is particularly important. We studied RFID survivability and the significant factors that affect survivability in the military domain. The US military applies RFID in several fields including the supply chain, asset management, maintenance, and healthcare. RFID plays a critical role in the acquisition, storage, distribution, disposition, and maintenance of military supplies. We formally defined the concept of RFID survivability and described the states of an RFID system under attack in terms of survivability. We showed how

the system could possibly transition from a survivable state to a non-survivable state. Based on the analysis of military RFID applications and the system survivability transition model, a theoretical survivability impact model has been developed which includes the following critical factors affecting RFID survivability in a military domain: adversarial threats and RFID security, RFID component/network robustness, RFID interoperability, and functional failure recovery.

A model has been developed to specify survivability requirements in order to make an RFID system more robust and resilient. Threat modeling provides insights into the possible threats to an RFID system. Risk analysis identifies those devastating attacks which could render an RFID system non-survivable. System survivability requirements are specified based on those identified attacks. These survivability requirements, if implemented, can significantly improve the RFID system's survivability. RFID developers and system designers can incorporate those survivability requirements into their software and hardware development processes. Classifying the requirements into major categories not only provides a systematic way to organize survivability requirements from users' perspective but also provides the guidelines for the system vendors to develop the most effective security/survivability mechanisms to satisfy the requirements as specified by the users in each category.

In an RFID enabled system, tags often need to change hands from one owner to another during their lifetime. We investigated securely transferring the ownership of a group of tags in one session. The protocol developed ensures new owner security, old owner security, and authorization recovery. It eliminates the possibility of dual ownership of an RFID tag in any time period during an ownership transfer process. The protocol resolves the windowing problem, where two entities possess the authentication information of the same tag for a certain period during the ownership transfer process.

We also developed a model for users to securely query an RFID system as a virtual database for any data that satisfies their search requirements. By viewing an RFID system as a virtual database, users can apply traditional database query languages (e.g., SQL) to seamlessly collect RFID data. There are three major layers in the system. The User Application Interface accepts SQL query statements from the higher-level applications and submits to the next lower layer. The Query Engine processes and transforms the high-level SQL statements to RFID specific query terms which are appropriate for the underlying sensor tags to process. The Query Implementation layer defines protocols and procedures for the RFID readers to submit the translated query terms to the underlying RFID network and for the sensor tags to process the query and return the partial results to the readers. Tag-reader communication represents the major entry point for potential attacks. The proposed protocol ensures reader-tag communication privacy and tag identity anonymity. It also resists tag impersonation and tag tracking attacks.

Effective tag search is both a necessary and invaluable tool in many RFID applications with a large number of tagged items. We proposed a set of protocols for an RFID reader to search for a particular tag based on its identity. A feature-based tag search protocol was further developed, which allows an RFID reader to search for a set of tags that satisfy certain criteria. System scalability is addressed as the number of tags in an RFID system becomes large. We analyzed the performance of the proposed protocols and their resistances to the security and privacy attacks as

identified in our threat model. Our study shows that the proposed protocols are both secure and private to a large set of malicious attacks.

Finally, we proposed a framework to apply RFID in securely tracing material flows in supply chains. Timely and accurate material flow control is critical to supply chain visibility and control. Secure tag-reader authentication protocols have been developed to ensure the authenticity of RFID tags and readers. Our approach requires constant time in the best case in RFID database key search in order to identify a legitimate tag, which makes real-time item identification possible and also increases RFID system scalability.

## 3. PERSONNEL SUPPORTED

*Faculty*:

- Yanjun Zuo (University of North Dakota)

*Graduate students* (the project supported the following students during their different states of studies at the University of North Dakota):

- Suhas Lande (M.S., Computer Science)
- Malvika Pimple (M.S., Computer Science)
- Liang Cheng (Ph.D., Computer Science)
- Abhilasha Bhatia (M.S., Computer Science)

*Research scholars* (the project partially supported the following research scholars during their short visits to the University of North Dakota):

- Lifeng Guo
- Haiyang Hu

## 4. PUBLICATIONS

This project has resulted in the following journal and conference publications as well as a book chapter. Currently, there are still a couple of papers under review, which have not been included in the list below.

1. Zuo, Y. (2012). "Survivability Experiment and Attack Characterization for RFID", *IEEE Transactions on Dependable and Secure Computing* (*TDSC*), (9)2, pp.289-302 (Impact factor: 2.09).
2. Julic, J. and Zuo, Y. (2012). "An RFID Survivability Impact Model in the Military Domain", *Proc. of 18th Americas Conference on Information Systems* (*AMCIS'12*), p. 9, August 8-12, Seattle, WA, USA.
3. Zuo, Y. (2012). "Incorporating Constraints to Software System Survivability Specification and Proof", *Proc. of 6th IEEE International Symposium on Theoretical Aspects of Software Engineering* (*TASE'12*), pp. 67-74, July 4-6, Beijing, China.

4. Zuo, Y. and Babin, J. (2012). "Searching for the Optimal Proof Schedule in a Proof-Carrying Survivability Paradigm – a Dynamic, Interactive Approach", *Proc. of 9th International Conference on Information Technology – New Generations* (*ITNG'12*), pp. 201-208, April 16-18, Las Vegas, USA.
5. Zuo, Y. and O'Keefe, T. (2011). "RFID-enabled Logistic Flow Tracing in Supply Chains: Communications, Protocol, and Security", *Proc. of the 2011 IEEE Global Communication Conference* (*GlobeCom'11*), p. 5, December 5-9, Houston, TX, USA (Acceptance rate: 36%).
6. Zuo, Y. and Lande, S. (2011). "A Logical Framework of Proof-Carrying Survivability", *Proc. of 10th IEEE International Conference on Trust, Security and Privacy in Computing and Communications* (*TrustCom'11*), pp. 472-481, November 16-18, Changsha, China (Acceptance rate: 26%).
7. Zuo, Y. (2011). "Designing Security Requirements – A Flexible, Balanced, and Threshold-based Approach", *Proc. of Americas Conference on Information Systems* (*AMCIS'11*), p. 10, August 4-7, Detroit, MI, USA.
8. Zuo, Y. (2011). "Automatic Proof of Survivability Compliance – Strategies and Techniques", *Proc. of 4th IEEE International Conference on Computer Science and Information Technology* (*ICCSIT'11*), p. 5, June 10-12, Chengdu, China.
9. Zuo, Y. (2010). "Survivable RFID Systems: Issues, Challenges, and Techniques", *IEEE Transactions on Systems*, *Man and Cybernetics* (*SMC*) *Part C*, *Special Issue on Availability*, *Reliability and Security*, 40(4), pp. 406-418 (Impact factor: 2.016).
10. Zuo, Y. (2010). "A Holistic Approach for Specification of Security Requirements in Ubiquitous Computing", *Proc. of 2010 International Conference on Information Systems* (*ICIS'10*), p. 14, December 13-16, St. Louis, MS, USA (Acceptance rate: 21%).
11. Lande, S., Zuo, Y. and Pimple, M. (2010). "Survivability Decision Model for Critical Information Systems Based on Bayesian Network", *Proc. of 5th Annual Symposium on Information Assurance* (*ASIA'10*), pp. 23-30, June 16-17, Albany, NY, USA.
12. Zuo, Y., Lande, S. and Pimple, M. (2010). "Analysis and Simulation of System Survivability", *Proc. of 7th International Conference on Information Technology: New Generations* (*ITNG'10*), pp. 36-41, April 12-14, IEEE Computer Society, Las Vegas, USA.
13. Zuo, Y. (2010). "RFID Survivability Quantification and Attack Modeling", *Proc. of the Third ACM Conference on Wireless Network Security* (*WiSec'10*), pp. 13-19, March 22-24, Hoboken, NJ, USA (Acceptance rate: 21%).
14. Zuo, Y., Hu, W. and O'Keefe, T. (2010). "Securely Querying Sensor Enabled RFID Virtual Databases", *Proc. of the 7th International Conference on Information Technology: New Generations* (*ITNG'10*), pp. 619-624, April 12-14, Las Vegas, USA.
15. Zuo, Y. (2010). "An Adaptation Based Survivability Framework for Mission Critical Systems", *Proc. of 5th International Conference on Information Warfare and Security* (*ICIW'10*), pp. 361-369, April 8-9, Dayton, Ohio, USA.

16. Zuo, Y. (2010). "Changing Hands Together: A Secure Group Ownership Transfer Protocol for RFID Tags", *Proc. of 43rd Hawaii International Conference on System Sciences* (*HICSS'10*), p. 10, IEEE Computer Society, January 5-8, Hawaii, USA.
17. Zuo, Y. (2010). "A Framework of Survivability Requirement Specification for Critical Information Systems", *Proc. of 43rd Hawaii International Conference on System Sciences* (*HICSS'10*), p.10, IEEE Computer Society, January 5-8, Hawaii, USA.
18. Zuo, Y. (2010). "Survivability in RFID Systems", Book Chapter, *Handheld Computing for Mobile Commerce: Applications, Concepts and Technologies*, ISBN: 1-615-20761-9, IGI Global.
19. Zuo, Y. (2009). "Secure and Private Search Protocols for RFID Systems", *Information Systems Frontiers* (*ISF*), *Special Issue on Advances in RFID Technology*, (12)5, pp. 507-519, Springer (Impact factor: 1.596).
20. Zuo, Y., Pimple, M. and Lande, S. (2009). "A Framework for RFID Survivability Requirement Analysis and Specification", *Proc. of International Joint Conference on Computing*, *Information and Systems Sciences and Engineering* (*CISSE'09*), p. 6, December 4-12, 2009, Bridgeport, CT, USA.
21. Zuo, Y. and Panda, B. (2009). "Unifying Strategies and Tactics: A Survivability Framework for Countering Cyber Attacks", *Proc. of IEEE International Conference on Intelligence and Security Informatics* (*ISI'09*), pp. 119-124, June 8-11, Dallas, Texas, USA.