Dependability and Security Models*

(Keynote Paper)

Kishor S. Trivedi, Dong Seong Kim, Arpan Roy
Department of Electrical and Computer Engineering
Duke University
Durham, NC, USA
{kst, dk76, ar2 }@ee.duke.edu

Deep Medhi
Computer Science & Electrical Engineering Department
University of Missouri-Kansas City
Kansas City, MO, USA
dmedhi@umkc.edu

Abstract— **There is a need to quantify system properties methodically. Dependability and security models have evolved nearly independently. Therefore, it is crucial to develop a classification of dependability and security models which can meet the requirement of professionals in both fault-tolerant computing and security community. In this paper, we present a new classification of dependability and security models. First we present the classification of threats and mitigations in systems and networks. And then we present several individual model types such as availability, confidentiality, integrity, performance, reliability, survivability, safety and maintainability. Finally we show that each model type can be combined and represented by one of the model representation techniques: combinatorial (such as reliability block diagrams (RBD), reliability graphs, fault trees, attack trees), state-space (continuous time Markov chains, stochastic Petri nets, fluid stochastic Petri nets, etc) and hierarchical (e.g., fault trees in the upper level and Markov chains in the lower level). We show case studies for each individual model types as well as composite model types.**

Keywords- ***availability; combinatorial model; dependability; hierarchical model; performance; reliability; security; state-space model; survivability.***

I. INTRODUCTION

The *term* dependability is commonly used by the fault tolerant and dependable computing community. Some researchers include security as one of the attributes of dependability. Security researchers consider confidentiality, integrity, and availability (and sometimes, non-repudiation) to represent the security status of the system and networks but it is lack of representing reliability and performance. Similarly several such dependability classifications exist but not all of them are useful to quantitatively assess the attributes of a wide variety of systems and networks. Therefore it is necessary to develop a new framework of dependability, security and survivability models to take into account attributes used in both reliability and security research communities. In this paper, we develop a novel classification of dependability, security and survivability models. We first present threats in networks and systems such as failures and attacks and then mitigations (countermeasures) against such threats. We divide model types into eight different categories: availability-type, confidentiality-type, integrity-type, performance-type, reliability-type, survivability-type, safety-type, and maintainability-type. Each of the model types can be used individually or it can be combined to represent composite model types. A stochastic model of any of these types can be constructed by three classes of model representation/analysis techniques: (i) combinatorial methods (such as reliability block diagram (RBD), fault tree, reliability graph, attack tree), (ii) state-space (continuous time Markov chain, stochastic Petri nets, fluid stochastic Petri nets etc) and (iii) hierarchical models (e.g., fault tree in the upper level and Markov chains in the lower level). We show practical examples for these model types using our model representation/analysis techniques. The remainder of this paper is organized as follows. The related work is presented in Section 2. The classification of threats and mitigation in networks and systems is presented in Section 3. Our classification of dependability models is presented in Section 4. Practical examples to illustrate our classification are presented in Section 5. Finally we conclude the paper in Section 6.

II. RELATED WORK

The term dependability has been assigned many different meanings in the literature. A 1988 survey of several definitions of computer-based system dependability resulted in the following summary: Dependability of a computer system may be defined as the justifiable confidence the manufacturer has that it will perform specified actions or deliver specified results in a trustworthy and timely manner [44]. Avizienis *et al.* [4] gave two alternative definitions for dependability as (i) "the ability to deliver service that can be justifiably trusted" (ii) "ability to avoid service failures that are more frequent and more severe than is acceptable". Levitt and Cheung [32] presented the common techniques used in fault-tolerance and security. They provided security counterparts to the most common fault-tolerance terms. Meadows [40] presented an outline of a fault model for security and showed how it could be applied to both fault tolerance and fault forecasting in computer security. Jonsson [27][28] proposed an integrated framework for security and dependability from the viewpoint

*This research was supported in part by the US National Science Foundation under Grant NSF-CNS-08-31325 and NSF-CNS-08-31090.

of behavioral and preventive terms. Meadows and Mclean [41] surveyed each part of the taxonomy for fault tolerance and described the research and practices in security that corresponded to it. Avizienis *et al.* [3] defined and summarized fundamental concepts of dependability. They presented the pathology of a failure; the relationship between faults, errors and failures and they compared the definitions of three widely known concepts: dependability, survivability, and trustworthiness. Avizienis *et al.* [4] refined the concept of dependability and security by emphasizing on security. They showed the relationship between dependability and security, both in the classic sense and as is relevant to telecommunications. The classical definition of dependability encompasses the attributes of reliability, availability, safety, integrity, and maintainability. The classical definition of security encompasses the attributes of confidentiality, integrity, non-repudiation, and availability. Nicol *et al.* [43] surveyed model-based techniques for quantitative evaluation. They presented measures of dependability and security and reviewed model representation/analysis techniques. Sallhammar *et al.* [47] proposed an approach to integrate security and dependability evaluation based on stochastic models. Hanmer *et al.* [24] explored the relationship between reliability engineering and security engineering for software products, especially concentrating on three areas; terminology, requirements, and common techniques. Soh and Dillon [53] present the notion of "fault tolerant security" and secure fault tolerance. More definitions of dependability can be found in [14][18]. Another related term is resiliency [47]. It is a combination of trustworthiness (dependability, security, performability) and tolerance (survivability, disruption tolerance, and traffic tolerance).

III. A CLASSIFICATION OF THREATS AND MITIGATIONS

An extension of Laprie [4] classification of threats is shown in Figure 1. We classify faults, errors, failures, attacks and compromise of security and overload, as summarized in Figure 1. We see that performance impairments, faults, attacks and accidents are threats to dependability and security. These may give rise to errors, intrusions, and overloaded situations, respectively. They may bring system failures, security compromises, or performance failures, respectively. Classification of mitigation techniques is shown in Figure 2 Figure 3 shows a small modification of the dependability and security tree from [4] as the basis of dependability models. We incorporate performance and survivability as new attributes to the dependability and security tree. We include accident as a new type of threat. Now we present each attribute of the dependability and security tree.

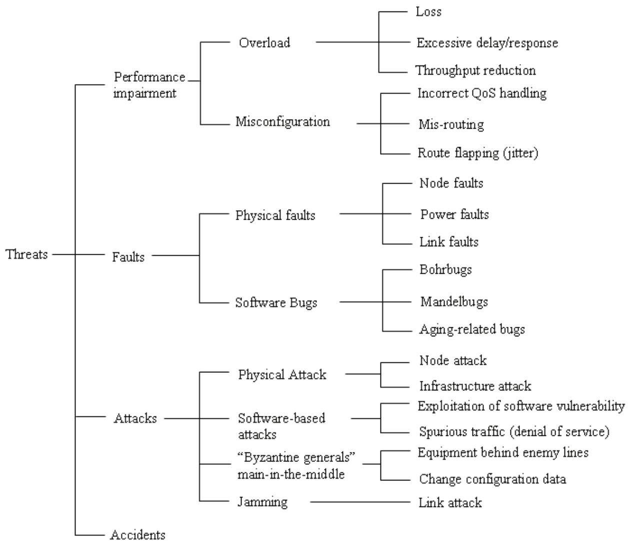


Figure 1. Classification of threats (failures, attacks, and accidents)

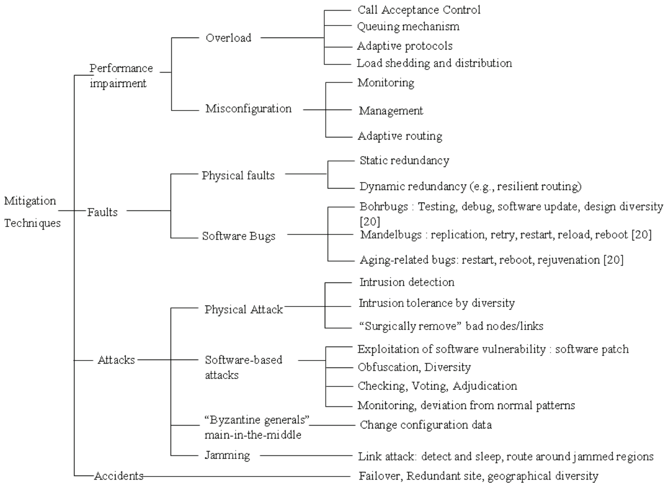

Mitigation Techniques
Performance impairment
Overload
Call Acceptance Control
Queuing mechanism
Adaptive protocols
Load shedding and distribution
Misconfiguration
Monitoring
Management
Adaptive routing
Faults
Physical faults
Static redundancy
Dynamic redundancy (e.g., resilient routing)
Software Bugs
Bohrbugs : Testing, debug, software update, design diversity [20]
Mandelbugs : replication, retry, restart, reload, reboot [20]
Aging-related bugs: restart, reboot, rejuvenation [20]
Attacks
Physical Attack
Intrusion detection
Intrusion tolerance by diversity
"Surgically remove" bad nodes/links
Software-based attacks
Exploitation of software vulnerability : software patch
Obfuscation, Diversity
Checking, Voting, Adjudication
Monitoring, deviation from normal patterns
"Byzantine generals" main-in-the-middle
Change configuration data
Jamming
Link attack: detect and sleep, route around jammed regions
Accidents
Failover, Redundant site, geographical diversity

Figure 2. Classification of mitigation techniques (countermeasures)

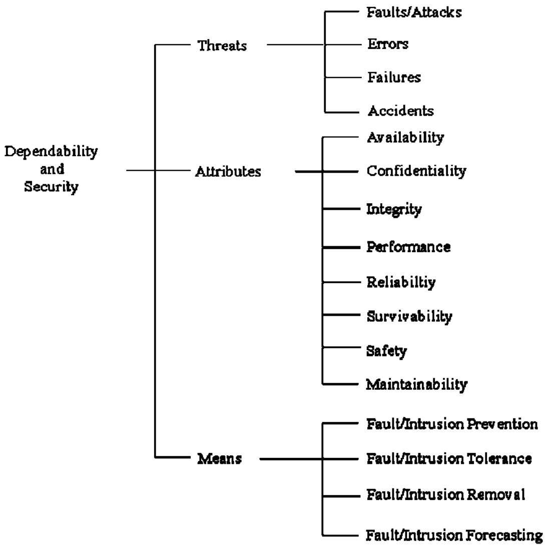


Figure 3. A dependability and security tree

- Availability (A): the ability of the system to perform its slated function at a specific instant of time or over a stated period of time. It is generally expressed in the form of a ratio of the units of time when service was available and the agreed service period.
- Confidentiality (C): the ability of the computing system to prevent disclosure of information to unauthorized parties. To ensure that information is accessible only to those authorized to have access.
- Integrity (I): the ability of the computer system to prevent unauthorized modification or deletion. In cryptography and information security in general, integrity refers to the validity of data.
- Performance (P): the degree to which the system or component accomplishes its designated functions within given constraints, such as speed, accuracy and memory usage. Performance can be considered as an attribute of dependability, although the classical literature does not include it in their definition [4]. ISO standard includes performance into a general definition of dependability [45].
- Reliability (R): the probability that a system performs a specified service throughout a specified interval of time [43]. The probability that the system has not failed once since it started service. It is a measure of the continuity of service. Repair after individual component failures are admitted but repair from a system failure is not permitted while computing reliability. Thus a state space type reliability model must have one or more absorbing states. By contrast an availability model will have no absorbing states.
- Survivability (S): the capability of a system to fulfill its mission, in a timely manner, in the presence of attacks, failures, or accidents, in [17]. Survivability is an important attribute in both the dependability domain and the security domain. Various mathematical definitions of survivability have been proposed in [25][33][34][35]. According to T1A1.2 (Network Survivability Performance) working group's definition [2], survivability depicts the time-varying system behavior after a failure, attack or accident occurs.
- Safety (SF): the capability of the system to avoid catastrophic consequences on the user(s) and the environment.
- Maintainability (M): the ability of the system to undergo modifications and repairs.

We present the new classification of dependability and security models in the next section.

IV. A CLASSIFICATION OF DEPENDABILITY AND SECURITY MODELS

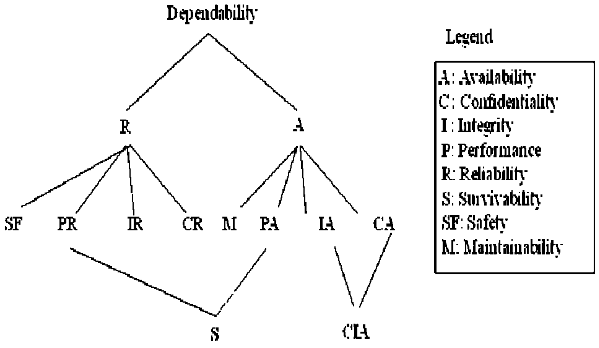


Figure 4. A new classification of dependability model types

A. A new classification of model types.

Figure 4 shows our classification of model types. At the root of the tree a distinction is made between R-type and A-type models. With R-type models once the system fails (due to hardware faults, software bugs, cyber attacks and accidents) occurs there is no restoration allowed while in the A-type models restoration from system failure is included in the analysis. Reliability focuses on the event when an error becomes visible (as a failure) at the service interface. It is a probability of failure-free operation in a specified period of time in a specified environment. There are no levels of service quality covered by (typical) reliability and availability models per se Performance models can be viewed as elaborating UP states of a R-type or A-type model into various levels of service. Combining performance and failure/recovery we obtain performability models. By considering transient analysis of system performance immediately after the occurrence of a failure, an attack or an accident, we obtain S-type (Survivability) model. I-type models represent the degree to which information is correct and consistent. C-type models capture the probability that information is accessible only to those authorized to have access. C-type and I-type models can thus be viewed as further classification of down or failure states of an R-type or A-type model. SF-type model can be viewed as a refinement of the down states of an R-type model into safe and unsafe states. R-type models can be divided into SF (safety), PR (Performance and Reliability), IR (Integrity and Reliability), and CR (Confidentiality and Reliability) model types. If we consider transient behavior of PR-type immediately after the occurrence of one of the threats, PR-type model becomes an S-type model. M-type model can be seen as an elaboration of an A-type model where the focus is on different strategies of maintenance: corrective vs. preventive; time-based vs. condition-based preventive. A-type models can be divided into M, PA (Performance and Availability), IA (Integrity and Availability), and CA (Confidentiality and Availability) type models. If we consider transient performance of PA-type model immediately after the occurrence of one of threats, PA-type model becomes S-type. Either IA or CA-type model can be further generalized to CIA (Confidentiality, Integrity and Availability) type model by incorporating C-type model. Using the above categories of model types, we can address a number of technical metrics related to security and dependability together. The benefit is that it helps to understand and assess the impact on overall systems and networks in a systematic manner. The next section presents case studies of each model type and composite model types using model representation/analysis techniques.

B. Case studies

1) Reliability model type (R-type) [7][20][35]:

Ramasamy *et al.* [46] quantified the impact of virtualization on node reliability using reliability block diagram (RBD). Figure 5(a) shows architecture of non-virtualized node and Figure 5(b) shows its combinatorial model. Figure 5(c) shows architecture of node with two Virtual Machines (VMs) and its combinatorial model. Reliability of each model can be computed based on the models. The threats for virtualized and non-virtualized system are physical faults in HW and software Mandel bugs in OS, Application, Virtual Machine Monitor (VMM), and Virtual Machines (VMs). Applying redundant hardware against physical faults and reboot after a failure due to software bugs are the countermeasures employed.

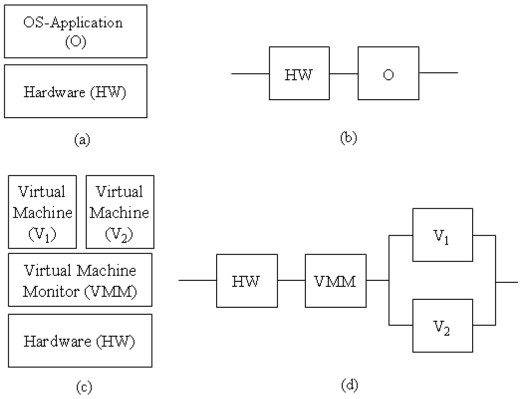


Figure 5. Non-virtualized node vs. virtualized node with two VMs**.**

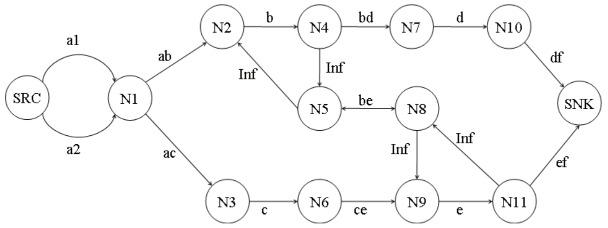


Figure 6. Reliability graph for a communication network example

Figure 6 shows reliability model of a communication network example using reliability graph [54]. The reliability of the same communication example can be represented using the fault tree [54] shown in Figure 7.

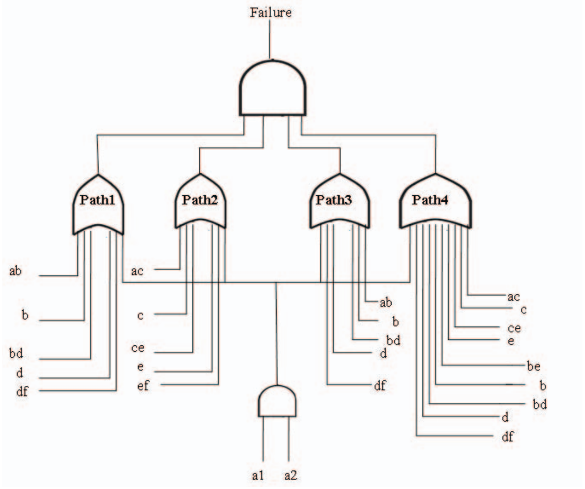


Figure 7. Fault tree model of a communication network example

2) Availability model type (A-type) [7][20][29][57]:

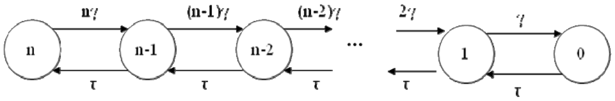


Figure 8. The availability model of a telecommunication system.

Figure 8 shows an availability model of a telecommunication switching system consisting of *n* trunks (or channels) with an infinite caller population [34]. Assume that the failure and repair times of each trunk are exponentially distributed with rates γ and τ, respectively. We also assume that a single repair facility is shared by all trunks in the system. The pure availability model of the system is a homogeneous continuous-time Markov chains (CTMC) as shown in Figure 8, where state *i* indicates that there are *i* non-failed trunks in the system. The threats in this example are trunks failures and mitigation is to repair trunks.

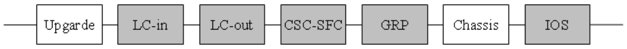


Figure 9. CISCO router availability model

Figure 9 shows availability model of CISCO GSR 12000 router [55]. The threats and countermeasures in this example are similar to those in Figure 5. This is a hierarchical analytic model in which upper model is a reliability block diagram (RBD) and lower level models are CTMCs. The block with grey color means there is CTMC submodel in the lower level. Such a hierarchical approach is also used to model availability of IBM SIP application [58] and of a virtualized system [29].

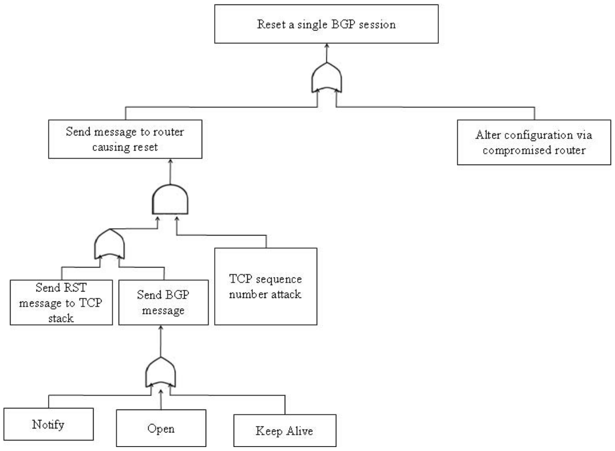

Reset a single BGP session
Send message to router causing reset
Alter configuration via compromised router
Send RST message to TCP stack
Send BGP message
TCP sequence number attack
Notify
Open
Keep Alive

Figure 10. Attack tree of reset of a single BGP session.

Figure 10 shows an attack tree that presents a reset of single BGP session [14]. In this attack, the attacker is trying to cause a current BGP session in the established state to reset. Such an attack could be launched over and over again to prevent two peers from reliably exchanging routing information so that this violates availability of BGP operation.

3) Performance model type (P-type) [20]

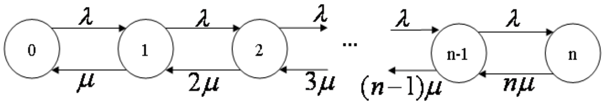


Figure 11. Performance model of a telecommunication system.

Figure 11 shows a performance model of a telecommunication system [34]. We revisit the example used in section 4.B.2. A call will be lost (referred to as blocking) when it finds all *n* trunks are busy upon its arrival. The call arrival process is assumed Poisson with rate λ. We assume call holding times are exponentially distributed with rate *μ*. Without considering link failures, the pure performance model is a homogeneous CTMC as shown in Figure 11, where *j* ongoing calls are present in state *j*. The threat in this example is possible performance impairment due to overload. If all *n* trunks are in use, the new call will be lost; this is named “loss” in the Figure 1. In order to minimize call loss, we can increase the number of channels, use call acceptance control and queue management as shown in Figure 2.

4) Performance and Reliability model type (PR-type)

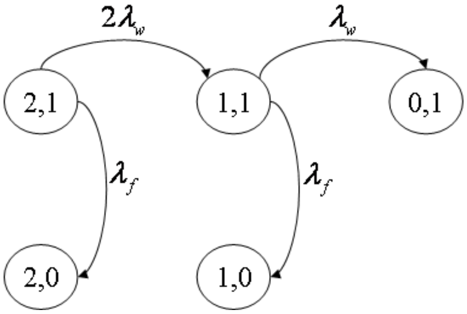


Figure 12. PR model of Workstation File server system

Figure 12 shows a simple PR-type model of Workstation File server System (WFS) example [56] in which there are two workstations and one file server. The system is operational as long as one of the workstations and the file server are operational. The state label *(i, j)* means that *i* workstations are still functional and *j* is 1 or 0 depending on whether the file-server is up or down. The Markov chain is basically R-type model because states *(0, 1)*, *(1, 0)* and *(2, 0)* are absorbing states. If reward rates (signifying the performance levels computed from a P-type model) are assigned to each of the states, CTMC model becomes a Markov reward model [56] and thus a hierarchical PR-type model. Work stations and file server can fail due to physical faults. Hardware redundancy is used a countermeasure.

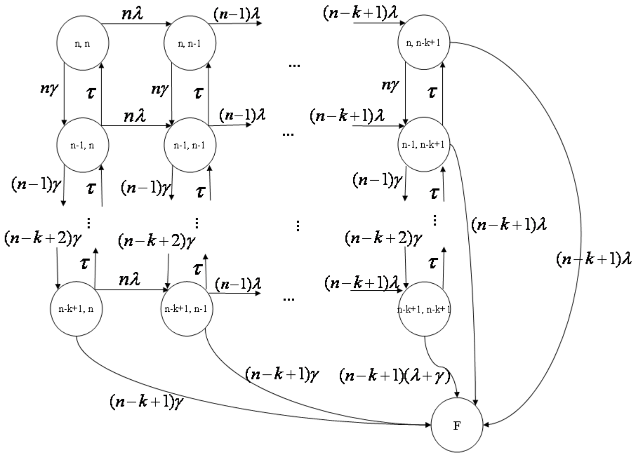


Figure 13. Confidentiality and Reliability-type model of sensor networks

5) *Confidentiality and Reliability model type (CR-type)*

Figure 13 shows a CR model type using a state-space approach. Suppose that there are *n* sensor nodes deployed in the field. Initially, *n* nodes are working properly. A sensor node can fail due to physical faults and/or software bugs thus hampering reliability. Each sensor node can fail with rate λ, and it can be repaired. Sensor node can be captured and the pairwise key(s) in the node can be acquired [61] (physical attacks - node attack in Figure 1) so that the attacker can violate confidentiality of the key(s) in the nodes. In key compromise, a sensor node loses its confidentiality with rate γ and it transits to its original state with rate τ if the compromised key is revoked and a new key is assigned. Once a sensor node fails, the sensor node can be compromised. After *k* (where, *k* <= *n*) number of sensor nodes fail, or k sensor nodes are compromised, no further repair/revocation is attempted and networks enter the failure state (F). In this way, confidentiality and reliability model types can be combined.

6) *Integrity and Reliability model type (IR-type)*

IR-model type using state space approach can be developed using the same model (Figure 13). Reliability model for sensor networks shows the same behavior as shown in Figure 13. In order to violate the integrity, the attacker can manipulate or change the data in the sensor node once he has captured sensor node [61]. We assume that once integrity of *k* sensor nodes is violated, we consider it as failure of the network.

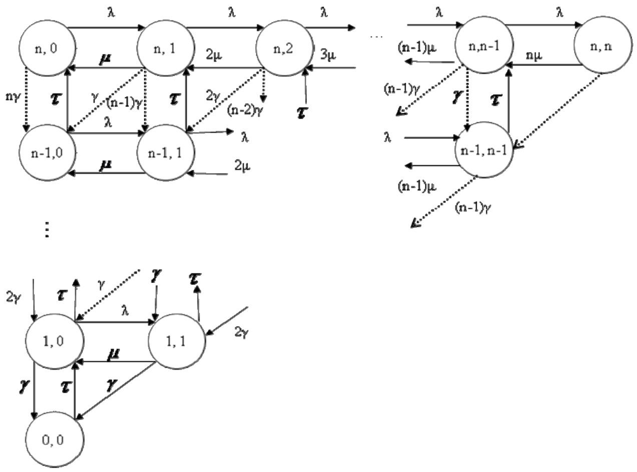


Figure 14. Composite Markov model for performance and availability

7) *Performance and Availability model type [39][51]*

Figure 14 shows PA model type of a telecommunication system. The model is a composite model which combines A-type (Figure 8) and P-type (Figure 11) model. This model incorporates both threats such as performance impairment and hardware failures. A hierarchical version of this model can be found in [30].

8) *Integrity and Availability model type (IA type)*

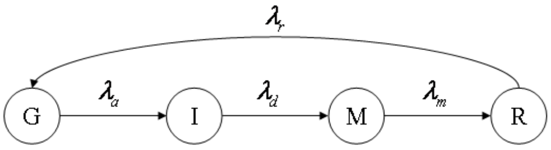
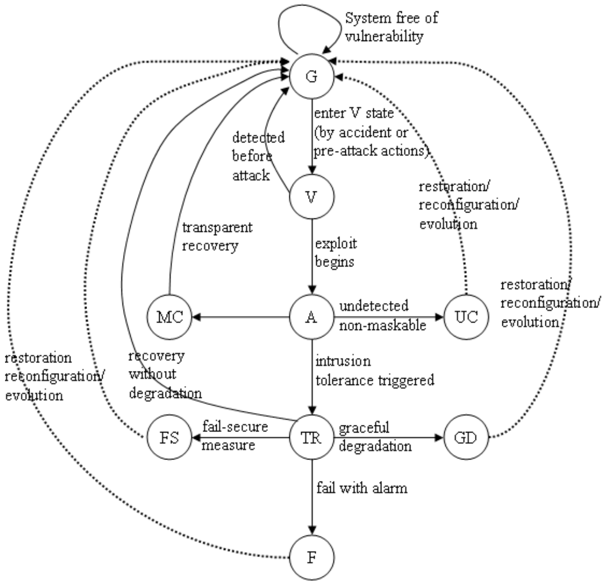


Figure 15. Markov model of an Intrusion tolerant database system

In [62], integrity and availability are proposed as quantitative measures to characterize the capability of a resilient database system surviving intrusions. The threat in this example can be software-based attacks (exploitation of software vulnerability) and its mitigation can be a software patch. The basic state transition model is a CTMC as shown in Figure 15 where G – good state, I- infected state, M- Malicious state and R- Repair state. The basic attributes associated with a database item were cleanliness and accessibility. Fraction of time a particular database item is clean signifies the integrity of that item and fraction of time a clean database item is accessible signifies the availability of the system. If a database item is 'dirty' and if a clean database item cannot be accessed, both represent loss of service to the user. So this is a classic example where integrity and availability model type together define the uptime of the system.

9) *Confidentiality and Availability model type (CA-type)*


System free of vulnerability
G
enter V state (by accident or pre-attack actions)
detected before attack
V
restoration/ reconfiguration/ evolution
transparent recovery
exploit begins
restoration/ reconfiguration/ evolution
MC
A
undetected non-maskable
UC
restoration reconfiguration/ evolution
recovery without degradation
intrusion tolerance triggered
FS
fail-secure measure
TR
graceful degradation
GD
fail with alarm
F

G	good state TR	TR	triage state
V	vulnerable state	FS	fail-secure state
A	active attack state	GD	graceful degradation state
MC	marked compromised state	F	failed state
UC	undetected compromised state		

Figure 16. Semi-Markov model for SITAR architecture

CA type model can be formalized using the model as shown in Figure 16 [21]. The model has taken instances of various forms of attacks and shown how and why each of the metrics -confidentiality and integrity can be computed. The threats are software-based attacks and accidents. The mitigation for software-based attacks is to use software patch. The mitigation for accident is to use failover, redundant site and geographical diversity. Availability is an appropriate measure for the compromise of data integrity and Denial of Service (DoS) attacks. It should be pointed out that in the case of DoS attacks which are aimed at disrupting normal services by consuming large amounts of service resource, states MC and FS do not make sense. Thus, it is not possible to mask a DoS attack by using redundancy. Also, intentionally making the system to stop functioning, i.e., bringing it to the FS will accomplish the goal of a DoS attack. Therefore, the states MC and FS will not be part of the state diagram describing DoS attacks. It follows that for the DoS attacks the system availability reduces to

$$A_{DoS} = 1 - \pi_F - \pi_{UC} \tag{1}$$

(where F: failed state and UC: undetected compromised state)

On the other hand, Microsoft IIS 4.0 suffered from the so-called ASP vulnerability as documented in the Bugtraq ID 1002. Exploitation of this vulnerability allows an attacker to traverse the entire web server file system, thus compromising confidentiality. Therefore, in the context of this attack, states UC and F are identified with the loss of confidentiality. Similarly, if the well-known *Code-Red* worm is modified to inject a piece of code into a vulnerable IIS server to browse unauthorized files, states UC and F will imply loss of confidentiality. Therefore, the steady-state confidentiality measure can then be computed as

$$C_{ASP} = 1 - \pi_F - \pi_{UC} \tag{2}$$

10) *Confidentiality, Integrity, Availability model (CIA-type)*

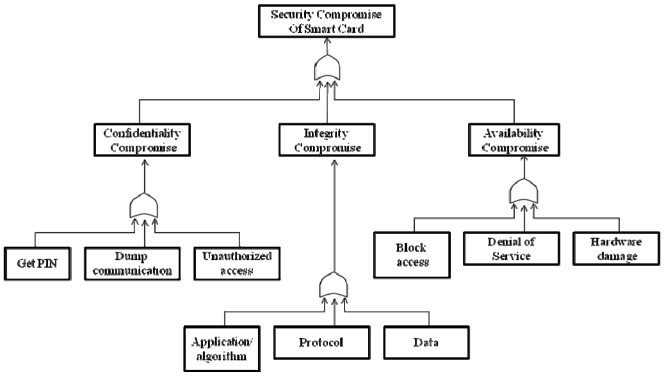


Figure 17. Attack tree for smart card

From the viewpoint of Ellison *et al.* [17], security can be defined as the combination of availability, confidentiality, and integrity and focuses on "*recognition of attacks*" and "*resistance of attacks*". Thus confidentiality, integrity and availability are by far the three metrics that are foremost in importance in the field of stochastic modeling of security. In the face of attacks, system behavior can be mapped into the state space model shown in Figure 16 and measures of interest are calculated by solving the model as a Semi-Markov Process (SMP) [21]. Figure 17 shows an attack tree for smart card. It is an example of CIA type combinatorial model. As shown in the figure, confidentiality of the smart card is compromised by stealing its PIN, sniffing and unauthorized access. Integrity is generally violated when the attacker can exploit a badly written protocol or an unsecure application or use of inefficient cryptographic technique on the data. Similarly availability is compromised by blocking PIN access, denial of service and hardware damage.

11) *Survivability (S-type)*

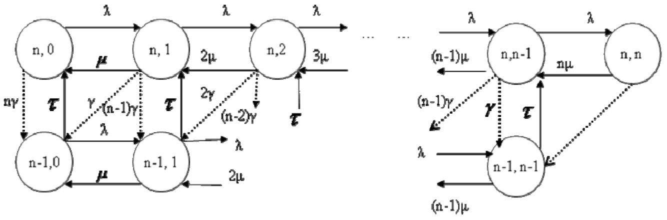


Figure 18. Survivability model type based on T1A1.2 definition.

Figure 18 shows a survivability model type using state-space model [34]. The threats to this model are to hardware and software faults, software-based attacks and/or accidents. According to the T1A1.2 survivability definition, it indicates the time dependent characteristics of the system behavior. If we only consider a single failure, we can truncate un-necessary states shown in Figure 14 resulting in the CTMC model shown in Figure 18. We force all the failure transitions (faults, attacks, and accidents in threat classification tree in Section 3A) from the first row to the second row; these transitions are marked with dotted arcs. In this way, we can study the transient system performance given that failures have occurred. Survivable systems should not only be able to survive faults, intrusions but also man-made accidents, natural disasters and terrorist attacks [25]. Fault tolerance does not (normally) consider malicious attacks (though Intrusion Tolerance does) and natural disasters. The system designer needs to follow three survivability design principles: decentralization which involves providing service without reliance on a common node in the architecture, redundancy i.e., providing service by switching (failing) over workload of the affected node(s) or link(s) to standby (backup) node(s) or link(s) and geographic diversity that is essential in survivable systems in order to avoid vulnerabilities to massive attacks or disasters by the placement of standby nodes or links outside of the expected radius of damage of related nodes or links.

12) Maintainability Model type (M-type) [8][9]

Figure 19 and 20 show Maintainability model types. The threats to this system are hardware faults. Corrective and preventive maintenance are employed as mitigation methods. Preventive maintenance is aimed at improving device availability or reducing repair costs when the device is in deterioration phase. The same models can be adapted to the software context where threats are aging-related software bugs. Corrective action is then a reboot of the operating systems while preventive action is known as software rejuvenation [5][60]. Basic state diagram of a system under time-based preventive maintenance is shown in Figure 19 [6][56] where there are three states; PM (preventive maintenance), UP and DOWN. The only available state is the UP state. Transition times are assumed to be generally distributed. In [60], a software system with condition-based of preventive maintenance of two types – minimal maintenance and major maintenance (see Figure 20) are presented. The system is assumed to degrade in *k* stages. D_1 to D_k represent the successive stages through which the software degrades. D_0 is the robust operational state. F is the final failure state. A full reboot is required from the F state to bring the system back to the D_0 state. Now during the successive deterioration stages, the system performs inspection represented by the states I_0 to I_k. Depending on the current inspection state different forms of maintenance is carried out. If for I_i , i <= g no maintenance is performed. If g< i <b, a minimal maintenance is performed represented by the states m_{g+1} to m_b with generally distributed sojourn time. Finally if i > b, then a major maintenance is performed where the system is brought back to the robust state D_0. The model is a Markov regenerative process (MRGP).

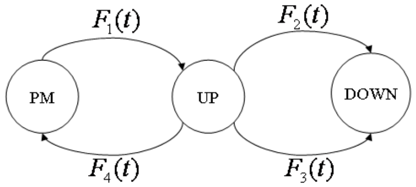


Figure 19. Semi-Markov model of time-based preventive maintenance.

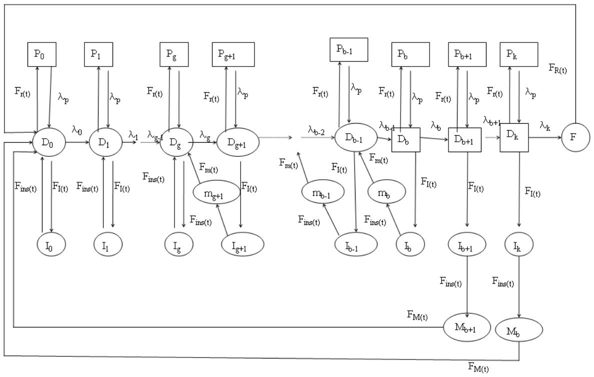


Figure 20. MRGP model of condition based preventive maintenance.

13) Safety Model (SF-type)

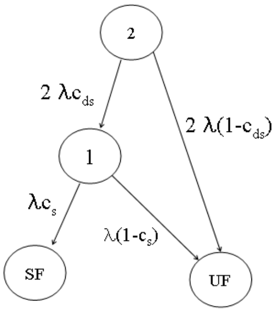


Figure 21. Safety model for a duplex system

A safety model for a duplex system is shown in Figure 21. The threats in this system are hardware faults. A system is said to undergo a safe failure (SF) when the failure is detected. If one processor fails and the failure is not detected then it is termed as a catastrophic (unsafe) failure. If the failure is detected the system is reconfigured for operation and it then enters a simplex state with one working processor. When this working processor fails, the failure is detected the system enters the safe absorbing state 'SF'. If the failure is not detected the system enters the unsafe failure state 'UF'. This model is used in applications like fault tolerant disk hardware architecture of RAID [11][56]. Undetected fault causes catastrophic failure whence corrupted data is passed to the user. A detected fault can be tolerated as the data lost by the failure of one disk can be reconstructed from the other operational disks using data and parity information. This is a reconfiguration phase. If data cannot be reconstructed or a second disk fails during reconstruction, system enters into an absorbing failed state called the data loss state representing a data loss situation. In [11], mathematical analysis is provided that helps in the calculation of conditional mean time to failure (MTTF) of system entering catastrophic failure and data loss states.

C. *Summary of model representation/analysis techniques*

TABLE I. DEPENDABILITY AND SECURITY MODEL REPRESENTATION

Model representation	**Dependability**	**Security**
Combinatorial models	RBD [20][46][54] Fault tree [20][54] Reliability graph [54]	Attack tree [14][30][42] Attack graph [50] A-Response graph [36]
State-space models	CTMC [8][20][34][56] SMP [6][9] MRGP [60] SRN [54]	SMP [18] CTMC [58] SITAR [21]
Hierarchical models	RBD+CTMC [26][37][55][57] fault tree+CTMC [29][58] SMP+CMTC [5]	N/A
fixed point iterative model	[23]	N/A
Simulation	[1]	[10][49]
Analytic and simulation	[13][25][59]	N/A
Hybrid model	HARP [16][19]	N/A

We summarize dependability and security model examples with respect to the modeling methods in Table 1. We have included fixed-point iterative, simulation and hybrid models as well. Combinatorial models, state-space model, and hierarchical models for dependability analysis are explained in earlier sections. In a hierarchical model, if the solution of a submodel is needed as an input parameter to another submodel than we say that the former submodel exports to the latter submodel. Such an export-import relationship can be depicted as an export-import graph [11][38]. In case, this graph is acyclic, it is easy to organize solution of submodels in such a way as to always have the needed input parameter values. In mode complex scenarios, however, the export-import graph may have cycles. In such cases, fixed point iteration can be used. Different sub-models may need to pass their solutions to other submodels as parameters via fixed-point iteration as in [23][52]. When the system is a non-Markovian (especially without regenerative structure), very few analytic methods are available. Discrete-event simulation can then be used to evaluate dependability and security of the system [59]. Simulative solution may also be needed in case of extremely large models that can otherwise be solvable using analytic-numeric methods [59]. For extremely difficult models analytic and simulation can be used in combination known as hybrid models [16][19].

V. CONCLUSIONS

In this paper, we have presented classification of threats and mitigations in systems and networks. We have presented a new classification of dependability and security models. We have presented individual dependability model type such as A, C, I, P, R, S, SF and M type model. We then showed that individual model types can be combined to form composite dependability model types. The dependability/security models can be represented as combinatorial models, state-space models, and hierarchical models. This has been described using case studies and illustrative examples. Due to space limitations we have chosen to exclude fixed-point iterative, simulation and hybrid models.

REFERENCES

[1] T. Angskun, et al., Reliability Analysis of Self-Healing Network using Discrete-Event Simulation, Proc. of CCGrid 2007.

[2] ANSI T1A1.2 Working Group on Network Survivability Performance, Technical Report on Enhanced Network Survivability Performance, ANSI, Tech. Rep. TR No. 68, 2001.

[3] A. Avizienis, J.-C. Laprie, B. Randell, Fundamental concepts of dependability, TR, LAAS-New Castle University-UCLA, 2001.

[4] A. Avizienis, J.-C. Laprie, B. Randell, C. Landwehr, Basic concepts and taxonomy of dependable and secure computing, *IEEE Trans. Dependable and Secure Computing*, 1(1) 2004.

[5] Y. Bao, X. Sun, K. S. Trivedi, A Workload-Based Analysis of Software Aging, and Rejuvenation, *IEEE Trans. Reliability* 54(3), 2005.

[6] D. Chen, K.S. Trivedi, "Analysis of Periodic Preventive Maintenance with General System Failure Distribution," Proc. of PRDC 2001.

[7] D. Chen, et al., "Reliability and availability analysis for the JPL Remote Exploration and Experimentation System," Proc. of DSN 2002.

[8] D. Chen, K. S. Trivedi, Closed-form analytical results for condition-based maintenance, *Reliability Engineering and System Safety* 76, 2002.

[9] D. Chen, K. S. Trivedi, Optimization for condition-based maintenance with semi-Markov decision process, *Reliability Engineering and System Safety* 90, 2005.

[10] S. D. Chi, et al., Network security modeling and cyber attack simulation methodology, Proc. of ACISP 2001.

[11] G. Ciardo, K. S. Trivedi, A Decomposition Approach for Stochastic Reward Net Models, Perform. Eval., 18(1), 1993.

[12] H. Choi, W Wang, K. S. Trivedi, "Analysis of conditional MTTF of fault tolerant system", *Microelectron and Reliability* 38,(3), 1998.

[13] P. K. Choudhary, B. B. Madan, K. S. Trivedi, "Modeling and Simulation of Integrated Voice/Data Cellular Communication with Generally Distributed Delay For End Voice Calls," Proc. of WSC 2005.

[14] S. Convey, D. Cook, M. Franz, An Attack Tree for the Border Gateway Protocol, 2003 : http://tools.ietf.org/html/draft-ietf-rpsec-bgpattack-00

[15] Dependability, http://www.cs.cornell.edu/Projects/secft/

[16] J. B. Dugan, K. S. Trivedi, Coverage Modeling for Dependability Analysis of Fault-Tolerant Systems, *IEEE Trans. Computer*, 38(6), 1989.

[17] R. J. Ellison, D. A. Fisher, R. C. Linger, H. F. Lipson, T. Longstaff, N. R. Mead. Survivable Network Systems: An Emerging Discipline. Technical Report CMU/SEI-97-TR-013, 1997.

[18] I. Eusgeld, F. C. Freiling, and R. Reussner (Eds.), Dependability Metrics, *LNCS* 4909, 2008.

[19] R. Geist, K. S. Trivedi, Reliability Estimation of Fault-Tolerant Systems: Tools and Techniques, *IEEE Trans. Computer*, 23(7), 1990.

[20] K. Goseva-Popstojanov, K. S. Trivedi, Stochastic Modeling Formalisms for Dependability, Performance and Performability, *LNCS* 1769, 2000

[21] K. Goseva-Popstojanova, F. Wang, R. Wang, F. Gong, K. Vaidyanathan, K. Trivedi, B. Muthusamy, "Characterizing intrusion tolerant systems using a state transition model," Proc. of DARPA Information Survivability Conference & Exposition II, 2001.

[22] M. Grottke, K. Trivedi, Fighting Bugs: Remove, Retry, Replicate, and Rejuvenate. *IEEE Trans. Computer* 40(2), 2007

[23] G. Haring, R. Marie, R. Puigjaner, K. S. Trivedi, Loss Formulae and Their Application to Optimization for Cellular Networks, *IEEE Trans. Veh. Technology*, 50, 2001.

[24] R. S. Hanmer, D. T. McBride, V. B. Mendiratta, Comparing Reliability and Security: Concepts, Requirements, and Techniques, *Bell Labs. Technical Journal*, 12(3), 2007.

[25] P. E. Heegard, K. S. Trivedi, Network survivability modeling, *Computer Networks* 53(8), 2009.

[26] O. C. Ibe, R. C. Howe, K. S. Trivedi, Approximate Availability Analysis of VAXcluster Systems, *IEEE Trans. Reliability* 38(1), 1989.

[27] E. Jonsson, L. Strömberg, S. Lindskog, "On the functional relation between security and dependability impairments," Proc. of NSPW 1999.

[28] E. Jonsson, "Towards an integrated conceptual model of security and dependability," Proc. of ARES 2006.

[29] D. S. Kim, F. Machida, K. S. Trivedi, "Availability Modeling and Analysis of a Virtualized System," Proc. of PRDC 2009.

[30] Z. Kincses, "Attack tree of smart cards", Technical Report, 2007.

[31] M. Lanus, L. Yin, K. S. Trivedi, Hierarchical composition and aggregation of state-based availability and performability models. *IEEE Trans. Reliability*, 52(1), 2003.

[32] N. Levitt, S. Cheung, "Common Techniques in Fault-Tolerance and Security," Proc. of DCCA 1994.

[33] H. F. Lipson, D. A. Fisher, Survivability—a new technical and business perspective on security, Proc. of NSPW 1999.

[34] Y. Liu and K. S. Trivedi, Survivability Quantification: The Analytical Modeling Approach, *Int. J. Performability Engineering*, 2(1) 2006.

[35] Y. Liu, Survivability of Networked Systems, PhD Dissertation, Duke University, 2008.

[36] B. B. Madan , K. S. Trivedi, Security modeling and quantification of intrusion tolerant systems using attack-response graph, *J. High Speed Networks*, 13(4), 2004

[37] M. Malhotra, K. S. Trivedi, "Reliability Modeling of Disk Array Systems," Proc. of TOOLS 1992.

[38] M. Malhotra, K. S. Trivedi, "A Methodology for Formal Expression of Hierarchy in Model Solution,"" Proc. of PNPM, 1993.

[39] Y. Ma, J. J. Han, K. S. Trivedi, Composite Performance and Availability Analysis of Communications Networks: A Comparison of Exact and Approximate Approaches, *IEEE Trans. Vehi. Technology*, 50(5), 2001.

[40] C. Meadows, "Applying the dependability paradigm to computer security," Proc. of NSPW 1995.

[41] C. Meadows, J. McLean, "Security and dependability: then and now," Proc. Computer Security, Dependability and Assurance: From Needs to Solutions, 1998.

[42] A. P. Moore et al., Attack Modeling for Information Security and Survivability, CMU TR, 2001.

[43] D. M. Nicol, W. H. Sanders, K. S. Trivedi, Model-Based Evaluation: From Dependability to Security, *IEEE Trans. Dependable and Secure Computing*, 1(1), 2004.

[44] B. Parhami, From defects to failures: a view of dependable computing, *Computer Architecture News* 16(4), 1988.

[45] Quality Concepts and Terminology, part 1: Generic Terms and Definitions. Document ISO/TC 176/SC 1 N 93, 1992.

[46] H. V. Ramasamy and M. Schunter, Architecting Dependable Systems Using Virtualization, In Workshop on DSN-2007.

[47] ResiliNets Wiki, Available at:

https://wiki.ittc.ku.edu/resilinets_wiki/index.php/Main_Page

[48] K. Sallhammar, B. E. Helvik, S. J. Knapskog, "Towards a stochastic model for integrated security and dependability evaluation," Proc. of ARES 2006.

[49] H. S. Seo and T. Cho, Modeling and Simulation for Detecting a Distributed Denial of Service Attack, Proc. of AI 2002.

[50] O. Sheyner et al., Automated Generation and Analysis of Attack Graphs, Proc. of IEEE SP 2002.

[51] R. M. Smith, K. S. Trivedi, A. V. Ramesh, Performability Analysis: Measures, an algorithm and a case study, *IEEE Trans. Computers* C-37(4), 1988

[52] W. E. Smith, K. S. Trivedi, L. A. Tomek, J. Ackaret, Availability analysis of blade server systems, IBM Systems J. 47(4), 2008.

[53] B. C. Soh, T. S. Dillon, "On Modelling and Analysis of Latency Problem in Fault-Tolerant Systems," Proc. 5th Int. GI/ITG/GMA Conference on Fault-Tolerant Computing Systems, Tests, Diagnosis, Fault Treatment 1991.

[54] K. S. Trivedi, S. Hunter, S. Garg, R. Fricks, "Reliability Analysis Techniques Explored Through a Communication Network Example," Proc. of CADTED 1996.

[55] K. S. Trivedi, Availability Analysis of Cisco GSR 12000 and Juniper M20/M40, Technical Report.

[56] K. S. Trivedi, Probability and Statistics with Reliability, Queuing, and Computer Science Applications. *John Wiley & Sons*, 2nd edition, 2001.

[57] K. S. Trivedi, R. Vasireddy, D. Trindade, S. Nathan, R. Castro, "Modeling High Availability Systems," Proc. of PRDC 2006.

[58] K. S. Trivedi, D. Wang, D. J. Hunt, A. Rindos, W. E. Smith, B. Vashaw, "Availability Modeling of SIP Protocol on IBM(c) WebSphere(c)," Proc. PRDC 2008.

[59] B. Tuffin, P. K. Choudhary, C. Hirel, K. S. Trivedi, "Simulation versus Analytic-Numeric Methods: a Petri Net Example," Proc. of the 2nd VALUETOOLS Conference 2007.

[60] K. Vaidyanathan, D. Selvamuthu, K. S. Trivedi, "Analysis of Inspection-Based Preventive Maintenance in Operational Software Systems," Proc. of SRDS 2002.

[61] J. P. Walters, Z. Liang, W. Shi and V. Chaudhary, "Wireless sensor network security: A survey", Journal of Security in Distributed, Grid, Mobile, and Pervasive Computing, 2007.

[62] H. Wang, P. Liu, "Modeling and evaluating the survivability of an intrusion tolerant database system," Proc. of ESORICS 2006.