MODELING AND ANALYSIS OF MULTISTAGE FAILURES OF A SYSTEM

Manoj Banik
Dept. of Modeling Simulation &
Visualization Engineering (MSVE),
Old Dominion University (ODU),
Norfolk, VA. USA
E-mail: mbani003@odu.edu

Bharat B. Madan
Dept. of Modeling Simulation &
Visualization Engineering (MSVE),
Old Dominion University (ODU),
Norfolk, VA. USA
E-mail: bmadan@odu.edu

ABSTRACT

This paper's focus is on modeling and analysis of security failures, resulting from cyber-attacks and subsequent repairs. An intruder exploits one or more vulnerabilities of a cyber system to compromise one (or more) of its security attributes. Since recovering from an attack can take unpredictably long time, cyber systems used in safety critical applications, need to be inherently capable of tolerating failures known as Intrusion Tolerant Systems (ITSs) which depend on incorporating redundancy in their design. Our analysis shows that plain replication, though effective against intrusions designed to compromise availability, can in fact be detrimental against confidentiality and integrity intrusions. This paper instead uses smart redundancy based on fragmentation, coding, dispersion and reassembly (FCDR), which is then used to develop multi-stage failures and repairs stochastic models. These models take the form of continuous time Markov chains (CTMSs), which are analyzed to quantify the effectiveness of smart redundancy in tolerating confidentiality, integrity and availability intrusions.

Keywords: intrusion tolerant systems, security, continuous time Markov chain, fragmentation, smart redundancy.

1 INTRODUCTION

Modeling of failures and subsequent repairs is an important part of analyzing and quantifying a system's performance. Cyber systems like other man-made systems experience failures, which generally occur randomly, at any time during its operation. Increasingly, cyber systems, made up of computers, software, sensors and data networks, are also being used to control and manage operation of physical systems in diverse application domains. Cyber-attacks seek to compromise a cyber-system's confidentiality (C), integrity (I) and availability (A) security attributes. An intruder exploits one or more vulnerabilities of a cyber system to compromise one (or more) of these security attributes, which is then used as a stepping stone to incrementally move towards the final objective of compromising systems' confidentiality, integrity or availability. Considering a system of n independent components and any component can fail at some arbitrary time with a non-zero probability. Modeling of failures and subsequent repairs is an important part of quantifying a system's performance metrics in terms of its reliability, availability and dependability. Enhancing these metrics has become an important requirement for safety critical systems, which can be met through robust design and by introducing redundancy in order to minimize failures, and when failures occur, resort to failure tolerance and continue to deliver services. We propose to model failures and repairs of such a system as multistage processes. Introduction of redundancy stochastic processes are natural choice to model system reliability and availability, due to randomness inherent in failure and repair processes. Increasingly, control and operation of physical systems is being deputed to cyber systems. Tight integration of physical and cyber systems has created a new class systems called

SpringSim-ANSS 2017, April 23-26, Virginia Beach, VA, USA

the Cyber Physical Systems (CPSs). Such CPSs have become vulnerable to additional sources of failures which can be attributed to deliberate and malicious actions cyber attackers (Cardenas, et al. 2009). Since CPSs are also widely deployed in our national infrastructure, failures in cyber systems used control and manage CPSs have acquired safety implications. This has raised the need for making such cyber systems to survive attacks by incorporating attack tolerance mechanisms. Attack modeling techniques such as Attack Trees (AT) (Schneier 1999) and Attacks Graphs (AG) (Roy, Kim and Trivedi 2012) have shown that an intruder exploits one or more vulnerabilities to compromise security attributes of an AT or AG node. The compromised nodes are then used as a stepping stone to incrementally move towards final objective of compromising a systems' confidentiality, integrity or availability. The first strategy involves use of intrusion prevention systems (e.g., strong security policies, firewalls, proxies, clean room software development, etc.), followed by intrusion detection/blocking systems (e.g., malware scanners, 3-legged firewalls, data analytics, etc.) and finally repair and recovery mechanisms. Unfortunately, it is not possible to guarantee zero failure probability of intrusion prevention, detection and recovery system. As a result, such a strategy by itself is not sufficient for ensuring security of cyber systems, particularly those used in critical applications.

The second strategy calls for architecting cyber systems that are inherently capable of tolerating failures resulting from security intrusions. Such systems, known as Intrusion Tolerant Systems (ITSs), are required to prevent, detect repair and recover from any damage caused by an intrusion within a reasonably small bounded time interval. In recent years, a good number of research articles have been published that describes intrusion tolerant architectures like SITAR (Wang, et al. 2001), SCIT (Huang 2002), Willow Architecture (Knight 2002). But all of these techniques are suffered from various limitations i.e., Denial of Service (DoS) attack on SITAR proxy servers will compromise the availability of the entire system, at SCIT (Huang 2002), there is no detection mechanism, so the system may continue to service while it was already compromised. All of these systems use naïve replication to achieve high throughput and reliability. Naïve replication however, lower the probability of availability compromise but increases the probability of confidentiality and integrity compromises. In this paper we propose fragmentation of data blocks or packets to eliminate single points of security failure. A data block or packet is fragmented into n fragments, which are augmented by k additional fragments using Erasure Codes (e.g., Reed-Solomon codes). Erasure Coding ensures that any n (out of $n + k$) fragments suffice to reproduce the original data block. Consequently, the cyber infrastructure can survive k availability attacks, $(n - 1)$ confidentiality and integrity attacks. The rest of the paper is organized as follows: Problem formulation is described at section 2. Smart redundant its and three models - availability, confidentiality and integrity using CTMC are shown section 3. Quantitative results and Conclusions are drawn at section 4 and 5.

2 PROBLEM FORMULATION

Basic replication of data in storage, processing and travelling over the network links is the basic strategy that can be used to achieve intrusion tolerance. As a result, failures of certain number of replicated subsystems or data blocks in an ITS do not prevent the system from functioning. Therefore, in an ITS, typically a failure resulting from an intrusion has to be dealt with first detecting the underlying intrusion using intrusion detection system (IDS) and after successful detection, recovery, cleanup and operations (henceforth referred to simply as a "repair process") to bring the system back to healthy state. We model an intrusion as a random event such that k^{th} intrusion is assumed to take random interval of time with $EXP(\lambda_k)$ distribution to achieve its objective of causing some kind of damage, and IDS is modeled as a Bernoulli process with P_{d_k} as the probability of successful detection of a k^{th} intrusion. Conversely, $\tilde{P}_{d_k}$ = 1 - P_{d_k} denotes the probability of an intrusion remaining undetected. On detecting a failure, the repair process starts instantaneously and the time taken to complete the repairs is modeled as continuous time

random process. So, we can model this multi-stage failure, detection and repair processes using continuous time Markov chain (Trivedi 2001).

The paper, considers an intrusion tolerant system where data replications are done through fragmentations and coding using the Reed-Solomon coding (Reed and Solomon 1960). A data block is first fragmented into n fragments and then k additional fragments are calculated from initial n fragments in such a way that any n of $(n + k)$ fragments are capable of reproducing the original data block. So the system can tolerate up to k intrusions. The attacks and repairs of the fragments are considered as a continuous time Markov chain which is shown in Figure 1. State s_i represents a state with i compromised fragments which means the system has been attacked i times successfully. We can assume that the system takes $t_{i,i+1}$ random time to make a transition from a state s_i to another state s_{i+1} and $t_{i,i-1}$ random time to transit from s_i to s_{i-1}. Furthermore, all such transition times are assumed to be exponentially distributed. Using the notation $Prob.(t_{i,j} \leq x) = p_{i,j}(x)$ and assuming all $p_{i,j}(x)$ are exponentially distributed, i.e., $p_{i,i+1}(x) = EXP(\lambda_i)$ and $p_{i,i-1}(x) = EXP(\mu_i)$. So, a successful attack on i^{th} state, make a transition to $(i + 1)^{\text{th}}$ state and similarly, a successful recovery on $(i + 1)^{\text{th}}$ state make a transition back to the i^{th} state. So the scenario is comparable with birth-death process where an attack/failure is considered as a birth and recovery from failures as a death process.

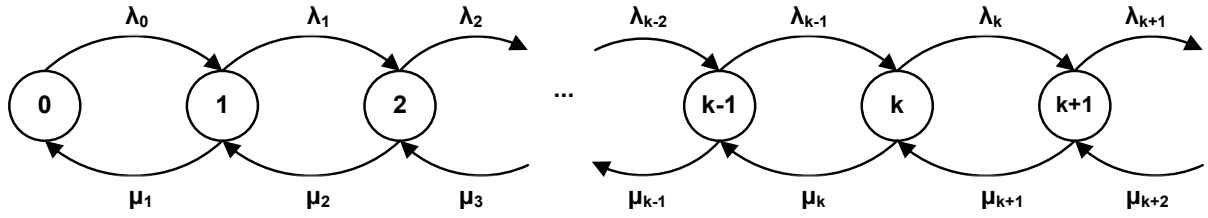

Figure 1: Continuous Time Markov Chain model of smart redundancy.

3 SMART REDUNDANT INTRUSION TOLERANT CYBER SYSTEMS

There are several security design principles that are useful for designing computer systems to survive from attacks (Avizienis, A. et al. 2004). Replication, for example, is a way to prevent a single-point of failure and ensure data availability. Well studied ITS architectures, e.g., SITAR (Wang 2001), SCITS (Huang 2002), MAFITA (Verissimo, et al. 2006) suggested high degree of replication to ensure service availability and reliability. GFS (Ghemawat, Gobioff and Leung 2003) and Hadoop (Hadoop project) employ high degree of replication to achieve high throughput, reliability and availability. However, high degree of naïve replication lower the probability of availability compromise but increases the probability of confidentiality and integrity compromises (shown in Table 1, section 4). To understand the whole risk, we can think of the following scenario. Let, N is the degree of redundancy for a system, p_a is the probability of a successful attack to compromise the availability of a single device. Then, the availability failure probability that N redundant systems will fail independently is given by, $P_A = \prod_{i=1}^{N} p_a = p_a^N$. Now making N large (N >> 1) we are able to keep P_A very small (<< 1), $0 \leq p_a \leq 1$. So, high degree of naïve replication works well against DoS attacks. From the confidentiality point of view, failure of any one of the N redundant copies of data will imply confidentiality failure of the entire storage system. Let p_c denotes the confidentiality failure probability of a single data copy. Assuming independent failure model, the confidentiality failure probability P_C of an N-redundancy system can be written as, $P_C = 1 - \prod_{i=1}^{N}(1 - p_c) = 1 - (1 - p_c)^N$ [p_c is the confidentiality failure probability of a single data fragment and P_C is the confidentiality failure probability of an N-redundancy system]. This expression suggest that, as N increases P_C also increases (close to 1) where $(0 \leq p_c \leq 1)$. Therefore, it follows that naïve replication

increases the probability of confidentiality compromises. Similar situation for the integrity performance also because the loss of security key for the hash function for a single copy means the loss for others copies too. So it is very clear that the high degree naïve replication is suitable for ensuring data availability but it increases the chances for confidentiality and integrity compromises.

In smart redundancy a data block is fragmented into $(n+k)$ fragments in such a way that any n fragments are capable to reproduce the original data block (Madan and Banik 2014). Therefore it follows that the availability failure will occur only if any $(k+1)$ out of $(n+k)$ fragments have their availability compromised. Each such event requires simultaneous failure of $(k+1)$ fragments. Consequently, overall availability failure probability will be given by, $P_A = 1 - \prod_{i=1}^{C(n+k,k+1)}\left(1-p^{k+1}\right) = 1-(1-p^{k+1})^{C(n+k,k+1)}$, where p is the failure probability of a single fragment and $C(x,y) = \binom{x}{y}$. Similar expression for the integrity failure probability (considering detectable attacks). Confidentiality failure will result if any n out of $(n+k)$ fragments have their confidentiality compromised. Each such event requires simultaneous failure of n fragments. Therefore, overall confidentiality failure probability will be given by $P_C = 1 - \prod_{i=1}^{C(n+k,n)}(1-p^n) = 1-(1-p^n)^{C(n+k,n)}$.

3.1 Availability Model

A smart redundant system can tolerate up to k intrusions capable of compromising its availability because availability of n out of $(n+k)$ fragments ensures data availability. Here, we are considering only non-stealthy (detectable) attacks because, the effects of damages are observables. Starting with a system in a pristine state s_0, in which all fragments are healthy and available, as time progresses, intruder is able to compromise the availability of one or more fragments. Let T_i (i = 1, 2, 3, …) denotes the random time at which the intruder manages to compromise i fragments. When an instruction is occurred, for each such intrusion, assuming that the system is able to detect intrusions instantaneously with probability 1 (subsequently, we relax this constraint by letting intrusion detection probability to be ≤ 1) and start a new mitigation process to recover affected fragment(s). Therefore, S tate s_i represents a s tate with i compromised fragments. We further assume that it takes $t_{i,i+1}$ random time to make transition from s_i to s_{i+1} and $t_{i,i-1}$ random time to transit from s_i to s_{i-1}. Furthermore, all such state transition times are assumed to be exponentially distributed. Using the notation $Prob.\left(t_{i,j} \le x\right) = p_{i,j}(x)$ and assuming all $p_{i,j}(x)$ are exponentially distributed, i.e., $p_{i,i+1}(x) = EXP(\lambda_i)$ and $p_{i,i-1}(x) = EXP(\mu_i)$. With these notations and assumptions, the resulting model turns out to be continuous time Markov chain (CTMC) shown in Figure 1. Consequently, in this model s_{k+1} represents the state in which systems is deemed to have become unavailable since the number of available good fragments are not enough to reproduce the original data block.

From steady-state solution of CTMC we know that, at any state the rates of flow into and out of the state are equal. So at state k the following relation will hold, $(\lambda_k + \mu_k)\,\pi_k = \lambda_{k-1}\pi_{k-1} + \mu_{k+1}\pi_{k+1}$. Where π_i Represents the steady state marginal probability that the system will be found at state i at some time T_i.

So, the steady state solution for the CTMC is given by the following set of equations.

$$\begin{bmatrix} -\lambda_0 & \mu_1 & 0 & \dots & 0 & 0 \\ \lambda_0 & -(\lambda_1+\mu_1) & \mu_2 & \dots & 0 & 0 \\ 0 & \lambda_1 & -(\lambda_2+\mu_2) & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & & -(\lambda_k+\mu_k) & \mu_{k+1} \\ 0 & 0 & 0 & \cdots & \lambda_k & -\mu_{k+1} \end{bmatrix} \begin{bmatrix} \pi_0 \\ \pi_1 \\ \pi_2 \\ \vdots \\ \pi_k \\ \pi_{k+1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \quad \dots\dots \quad (1.1)$$

Since the system will stay at any one of $(k+1)$ states at any given time, it must hold the following relationship,

$$\sum_{i=0}^{k+1} \pi_i = 1 \quad \ldots\ldots \quad (1.2)$$

Solving (1.1) and (1.2) yields,

$$\pi_i = \pi_0 \prod_{j=0}^{i} \left(\frac{\lambda_j}{\mu_{j+1}}\right) \quad \ldots\ldots \quad (1.3)$$

λ_j and μ_j are the rate of failures and recoveries respectively for the whole system. We make $\lambda_j = (n + k - j)\lambda$ and $\mu_j = j * \mu$ where λ & μ are the rate of failures and recoveries respectively for a single fragment. The values for λ & μ will depend on the environment where the system is working. Replacing λ_j and μ_j in equation (1.3) we will get the following one,

$$\pi_i = \pi_0 \frac{(n+k)!}{(n+k-i)!(i)!} \left(\frac{\lambda}{\mu}\right)^i \quad \ldots\ldots \quad (1.4)$$

Substituting (1.4) into (1.2) we can find,

$$\pi_0 = \frac{1}{1+\sum_{i=1}^{k+1} \frac{(n+k)!}{(n+k-i)!i!}\left(\frac{\lambda}{\mu}\right)^i} \quad \ldots\ldots \quad (1.5)$$

3.2 Confidentiality Model

Since n fragments are needed to reconstruct the original data, the system can survive up to $(n-1)$ confidentiality attacks. As for the confidentiality case, stealthy attacks are the only one that makes sense. So, the confidentiality model is very much similar to the availability model, only difference is with the state number where the confidentiality of the system has been compromised. For availability model the compromised state is s_{k+1} and in confidentiality model it will be the state s_n. When the system is at state s_n (n fragments are being compromised), we can say that the confidentiality has been compromised considering the fact that attacker will be able to reproduce the original data block. So, the steady state solution for the CTMC for confidentiality model can be obtained by carefully replacing (k+1) by n in the equations (1.2) & (1.5).

$$\sum_{i=0}^{n} \pi_i = 1 \quad \ldots\ldots \quad (1.6)$$

Combining (1.4) and (1.6) gives,

$$\pi_0 = \frac{1}{1+\sum_{i=1}^{n} \frac{(n+k)!}{(n+k-i)!i!}\left(\frac{\lambda}{\mu}\right)^i} \quad \ldots\ldots \quad (1.7)$$

The expression for π_i (for confidentiality model) remains same as availability model.

3.3 Integrity Model

The integrity attribute of security is concerned with preventing unauthorized modifications to data during processing, storage, and network transportation phases (Madan and Banik 2014). There are some standard techniques like Digital signatures (www.cgi.com) and secure hash functions (Kuwari 2010) that can be used for securing data integrity. When such techniques are being used, loss of integrity and loss of confidentiality get closely related since both depend on the strength of cryptographic keys and algorithms. When data is replicated in naive way, loss of integrity of any single copy amounts to the loss of overall system integrity because of compromise of the cryptographic keys used to compute secure hash values. Let p_i denotes the integrity failure probability of a single data copy. Assuming independent failure model (since fragments are under the control their respective storage systems/servers, so failures in these servers

are mutually independent), the overall integrity failure probability P_I of an N-redundancy system can be written as, $P_I = 1 - \prod_{i=1}^{N}(1 - p_i) = 1 - (1 - p_i)^N$. This expression suggests that, as N increases P_I also increases, where $(0 \leq p_i \leq 1)$.

For smart redundancy systems, loss of integrity (or any other security attribute) for a single fragment does not necessarily imply the loss of the integrity for the whole system. Because there is a high probability that during the reconstruction process, we could ended up excluding the integrity compromised fragment(s) (in case of stealthy attacks) in the bundle of n fragments. Majority voting process can also be applicable to deal with the integrity issues (Madan 2016). The following paragraph helps us finding the probability that a random bundle of n fragments out of $(n + k)$, contains one or more compromised one. When there are only one compromised fragment, the total number of bundles/combinations (n out of $n + k$) that include the bad fragments are $\binom{n+k-1}{n-1}$ [Explanation: We selected the bad one first and then we need to select any $(n - 1)$ out of $(n + k - 1)$]. So the probability that a bundle of n fragments ended up with the bad fragment is $\{\binom{n+k-1}{n-1} / \binom{n+k}{n}\}$, where $\binom{n+k}{n}$ are the possible ways to make a bundle of n fragments. When there are two compromised fragments, the total number of bundles/combinations (n out of $n + k$) that do not include any bad fragments are $\binom{n+k-2}{n}$ [Explanation: We selected n fragments out of all good fragments]. The total number of bundles/combinations that contain one or more bad fragments are $\{\binom{n+k}{n} - \binom{n+k-2}{n}\}$. The probability that a bundle of n fragments ended up with one or more bad fragment is $\{\binom{n+k}{n} - \binom{n+k-2}{n}\} / \binom{n+k}{n}$. So, when there are r compromised fragments, the probability that a bundle of n fragments ended up with one or more bad fragment is $\{\binom{n+k}{n} - \binom{n+k-r}{n}\} / \binom{n+k}{n}$ when $r <= k$, otherwise it will be 1 [because when r > k there are not enough healthy fragments remaining to construct the original data block]. The above discussion suggests that the probability of integrity failure could be lower in smart redundancy even there are some compromised fragments or the attacks are stealthy.

For non-stealthy (detectable) attacks, the integrity model is similar with the availability model. Because during reconstruction process, all compromised fragments can easily be identified (using hash function) and excluded in the bundle of n fragments. So, equations (1.4) & (1.5) can also be used to find the marginal probabilities for the compromised state and all other safe states for integrity model.

For stealthy attacks, the model needs to capture two distinct type of situations that interacts with each other. When an attacker succeeds in compromising a fragment, the compromised fragment belongs to one of two mutually exclusive classes – (1) compromised fragment is not part of the bundle selected for reassembly or (2) compromised fragment is part of the bundle selected for reassembly. This leads to 2-D CTMC model involving two interacting random processes. In this paper we have limited our discussion only on simple (1-D) CTMC model. So the integrity model is similar with the availability model.

4 QUANTITATIVE RESULTS

This section is divided into two parts. First part shows us a comparison between the naïve replication and smart redundancy system in terms of failure probability for different security attributes and in the second part, we will find the marginal probabilities for a steady state smart redundancy system for a given failure and recovery rate.

To compare both the systems, we assume some values for the parameters N, n, k & p to find the overall failure probability for a system for three security attributes A, C & I. Table 1 shows the comparison for

both the systems. All the calculations are done by MS Excel 2013. Table entries are found by the expressions shown at section Smart Redundancy Intrusion Tolerant Cyber Systems.

Table 1: Comparison of failure probabilities for various security attributes between naïve and smart redundancy systems (p denotes the failure probability for a single file/fragment for all security attributes).

		Naïve	Smart	Naïve	Smart
		N=5	n=6, k=6	N=10	n=6, k=12
p=0.06	Availability	7.776E-07	2.21709E-06	6.05E-13	9.51239E-13
	Confidentiality	0.266095978	4.31092E-05	0.4613849	0.000865747
	Integrity	0.266095978	2.21709E-06	0.4613849	9.51239E-13
p=0.1	Availability	0.00001	7.91969E-05	1.00E-10	8.57066E-10
	Confidentiality	0.40951	0.000923574	0.6513216	0.018392759
	Integrity	0.40951	7.91969E-05	0.6513216	8.57066E-10
Storage Requirement		5 times	2 times	10 times	3 times

From Table 1 it is clear that, though the smart redundancy has little higher failure probability for availability compared to naïve replication system (for lower values of n or k) but it has tremendous improvement for confidentiality and integrity security attributes. The naïve replication system provides lower failure probability only for the availability security attribute but it has very high failure probability for other two cases and every time the degree of replication increases, the failure probability for confidentiality and integrity goes up (we can see this in the Table I as N changed from 5 to 10). Smart redundancy maintains a very good balance on all three security attributes with less storage. Smart redundancy can even lower the failure probability further for all three security attributes by increasing the number of fragmentation (n) or by increasing the number of checksum fragments k. The only drawback for this system is, the fragmentation and reassembly process will slower the overall system a little bit.

The marginal probability for the states of a steady state smart redundant system has been found for different combinations of n, k, λ & μ which are shown in the following Figures 2a, 2b, 2c & 2d. The values are calculated from the expressions shown at section availability and confidentiality model using a C++ program. Graphs are drawn by MS Excel 2013.

From the graphs (Figures 2a, 2b, 2c & 2d), it is clear that when the repair rate μ is higher than the failure rate λ, the system stays good with the highest probability. The overall failure and repair rate depends on state, $\lambda_i = (n + k - i)\lambda$ and $\mu_i = i * \mu$, where λ and μ are depends of the system environment (more or less cyber-attacks) (www.forbes.com). But when the failure rate is higher than the recovery rate ($\lambda/\mu > 1$), then there is a high chance that the system will found in the compromised state (last state in the graph). In all the Figures from 2a to 2d, the last state is the compromised state and in all cases the system stays at last state or close to the last state (with highest marginal probability) when $\lambda/\mu > 1$. But when μ getting bigger than λ, the state with highest marginal probability moves to the left, which indicates the system stays in a good working state with highest marginal probability.

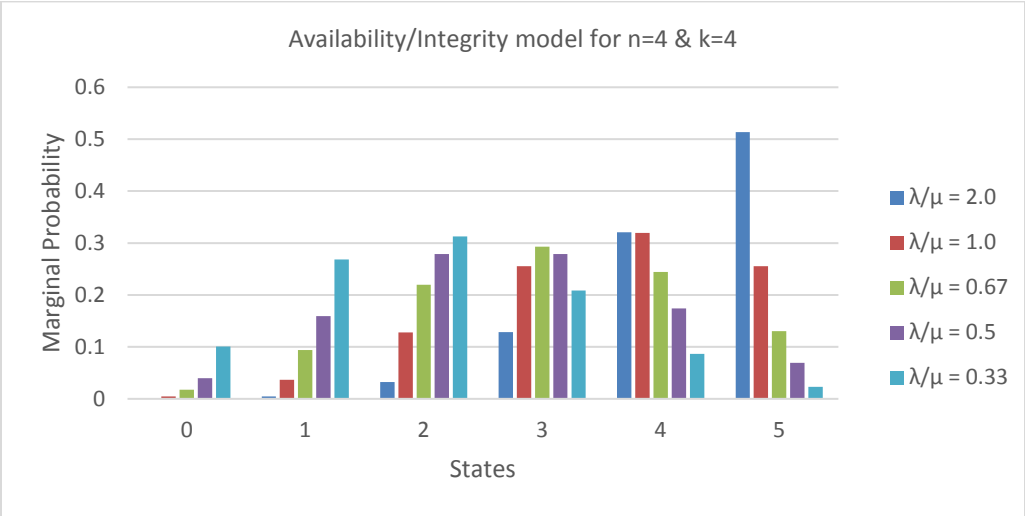

Figure 2a: Availability/Integrity model for n=4 & k=4.

In Figure 2a, when $\lambda/\mu = 2.0$ the state with highest marginal probability is found at 5 which is a compromised state. Because, at state 5 there are 5 damaged fragments and the system is not able to reconstruct the original data block. But, when $\lambda/\mu = 1.0$ the state with highest marginal probability is found at 4 which is a good state and the system is still available for service. The situation keeps improving when the repair rate getting higher compared to failure rate. The state with highest marginal probability can be found at 3 when $\lambda/\mu = 0.67$ and 2 when $\lambda/\mu = 0.5$ & 0.33. Similar situation for the remaining figures.

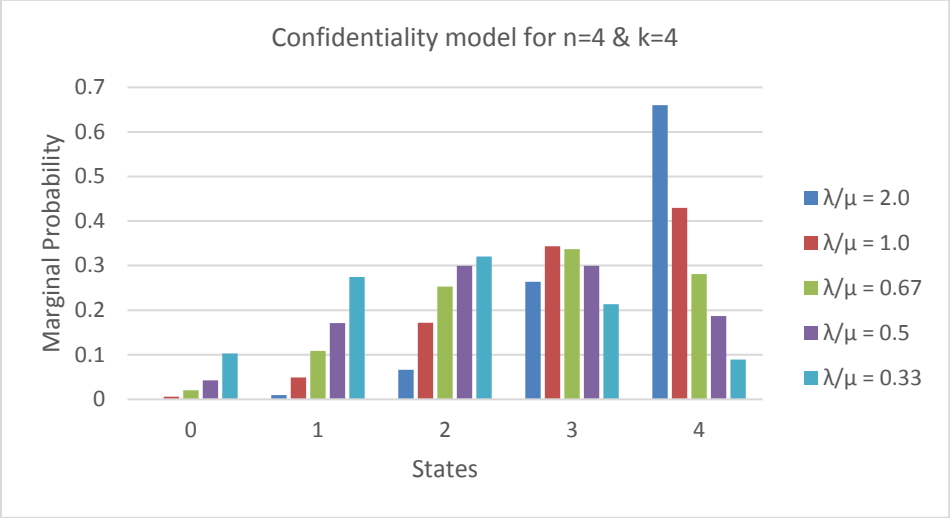

Figure 2b: Confidentiality model for n=4 & k=4.

In Figure 2b, the state with highest marginal probability is 4(compromised state) when $\lambda/\mu = 2.0$ & 1.0 but the system is available for service when $\lambda/\mu = 0.67$, $\lambda/\mu = 0.5$ & 0.33. In Figure 2c, the state with highest marginal probability is 8 when $\lambda/\mu = 2.0$, 6 when $\lambda/\mu = 1.0$, 5 when $\lambda/\mu = 0.67$, 4 when $\lambda/\mu = 0.5$ and 3 when $\lambda/\mu = 0.33$. In Figure 2d, the state with highest marginal probability is 8 when $\lambda/\mu = 2.0$ & 1.0, 6 when $\lambda/\mu = 0.67$, 5 when $\lambda/\mu = 0.5$ and 4 when $\lambda/\mu = 0.33$.

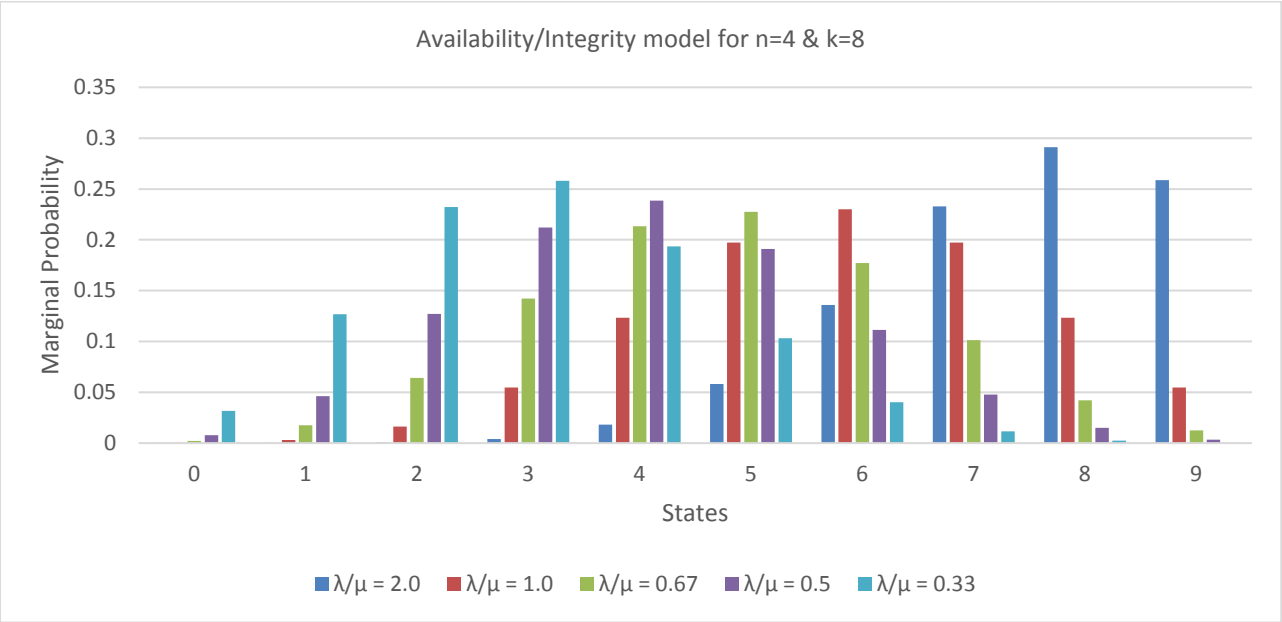

Figure 2c: Availability/Integrity model for n=4 & k=8.

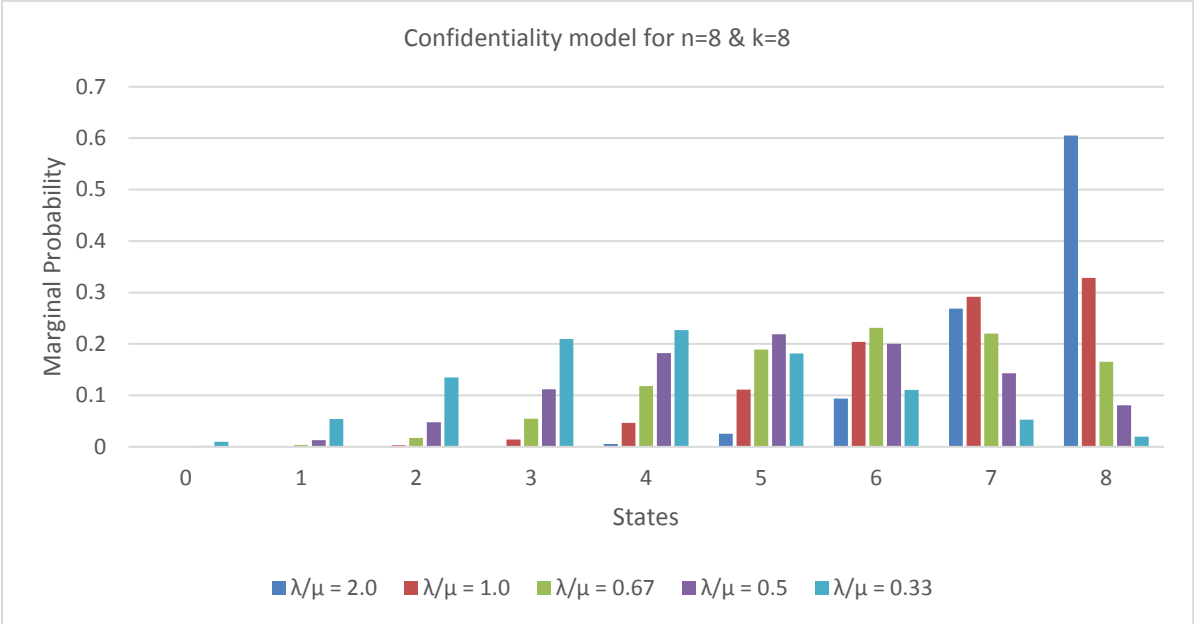

Figure 2d: Confidentiality model for n=8 & k=8.

5 CONCLUSIONS AND FUTURE WORK

The paper models failure and recovery processes of a multistage failures system as CTMC. The paper also shows that the smart redundancy concept could play an important role in the intrusion tolerant systems over naïve redundancy. From quantitative result it was found that a smart redundant system has

much less failure probability for three basic security attributes compared to naïve redundancy. The paper also shows that the system still remains available for service under attack as long as the repair rate is higher than the failure rate in smart redundancy. Paper covers both detectable and stealthy attacks for availability and confidentiality security compromises but for integrity security compromises, paper focuses only on detectable attacks. Authors want to explore stealthy attacks for integrity issues using 2-D CTMC model in future work. Authors also keep in mind to find the processing overhead (time required for fragmentation and reconstruction) in future work.

ACKNOWLEDGMENTS

This work is supported by the Department of Modeling Simulation & Visualization Engineering (MSVE), ODU.

REFERENCES

Ghemawat, S., H. Gobioff, and S.-T. Leung. 2003. “The Google file system”. *In 19th Symposium on Operating Systems Principles*. Lake George, NY. 29-43.

Trivedi, K. S. 2001. “Probability and Statistics with Reliability, Queuing, and Computer Science Applications”. *John Wiley and Sons*, New York.

Wang, F., et al. 2001. "SITAR: A scalable intrusion-tolerant architecture for distributed services". *Proceedings of the Second Annual IEEE Systems, Man, and Cybernetics Informations Assurance Workshop*, West Point, NY.

Huang, Y. and A. Sood. 2002. “Self-Cleansing Systems for Intrusion Containment” *Proceedings of Workshop on Self-Healing, Adaptive, and Self-Managed Systems (SHAMAN)*, New York City.

Verissimo, P. et al. 2006. “Intrusion-tolerant middleware: the road to automatic security”. *In: IEEE Secuirty & Privacy* 4.4.

Hadoop project. http://hadoop.apache.org.

Madan, B. B., M. Banik. 2014. “Attack tolerant architecture for big data file systems”. *ACM SIGMETRICS Perform. Eval. Rev*., 41 (4) , pp. 65–69.

Avizienis, A. et al. 2004. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing*, 1(1):11–32.

Schneier, B. 1999. "Attack Trees Modeling Security Threats", *Dr. Dobb's Journal*.

Roy, A., D. S. Kim and K. S. Trivedi. 2012, “Attack countermeasure trees (ACT): towards unifying the constructs of attack and defense trees”. *Security and Communication Networks* 5(8), 929–943.

Knight, J., D. Heimbigner and A. Wolf. 2002. “The Willow Architecture: Comprehensive Survivability for Large-Scale Distributed Applications”, *Intrusion Tolerance System Workshop, Supplemental International Conference on Dependable, System and Network.*

Public Key Encryption and Digital Signature: How do t hey work? http://www.cgi.com/files/white-papers/cgi_whpr_35_pki_e.pdf.

Kuwari, S. A., J. H. Davenport, R. J. Bradford. 2010. “Cryptographic Hash Functions: Recent Design Trends and Security Notations”. *In short paper proceedings of Inscrypt’10*. Science Press of China, pp- 133-150.

Reed, I. S., and G. Solomon. 1960. “Polynomial codes over certain finite fields”. *Journal of the Society for Industrial and Applied Mathematics*, 8:300– 304.

Madan, B. B., M. Banik. and B. C. Wu. 2016. “Intrusion Tolerant Cloud Storage System”, Submitted to the *IEEE Trans. Dependable and Secure Computing*. (ISI Rated).

Cardenas, A., et al. 2009. "Challenges for securing cyber physical systems." *Workshop on future directions in cyber-physical systems security*.

http://www.forbes.com/sites/stevemorgan/2016/05/13/list-of-the-5-most-cyber-attacked-industries/#3f35c37e3954.

AUTHOR BIOGRAPHIES

MANOJ BANIK is a PhD candidate in the department of Modeling, Simulation and Visualization Engineering (MSVE) at Old Dominion University (ODU), Norfolk, Virginia. He has started his PhD program at Fall 2013. He received his BS and MS in Computer Science and Engineering from Bangladesh University of Engineering and Technology (BUET) and United International University (UIU), Dhaka, Bangladesh, respectively. He was a faculty at the Department of Computer Science and Engineering at University of Asia Pacific (UAP) and Ahsanullah University of Science and Technology (AUST), Dhaka, Bangladesh. His research area includes Intrusion tolerant systems, Computer communications, and Neural networks.

BHARAT MADAN is an Professor in the department of Modeling, Simulation and Visualization Engineering (MSVE) at Old Dominion University (ODU), Norfolk, Virginia. Prior to this he was with the Applied Research Lab, Penn State University, where he headed the Distributed Systems Department. He has also held academic positions at the Indian Institute of Technology (New Delhi), Duke University, University of Delaware and the Naval Postgraduate School (Monterey, CA). His teaching and research interests are in computer and network systems security, attack tolerant survivable architectures, real-time systems security, sensor data fusion and autonomous system collaboration.