Communications of the IIMA

Volume 9 | Issue 3 Article 1

2009

Simulation of a Two-Category Secured Access Database

Marn Ling Shing
Taipei Municipal University of Education

Chen-Chi Shing
Radford University

Kuo Lane Chen
University of Southern Mississippi

Huei Lee
Eastern Michigan University

Simulation of a Two-Category Secured Access Database

Marn-Ling Shing
Taipei Municipal University of Education
TAIWAN
sshing@tmue.edu.tw

Chen-Chi Shing
Radford University
USA
cshing@radford.edu

Kuo Lane Chen
University of Southern Mississippi
USA
Kuo.chen@usm.edu

Huei Lee
Eastern Michigan University
USA
Huei.lee@emich.edu

ABSTRACT

Electronic commerce users continually access sensitive information on business databases through the Internet. In order to protect databases from unauthorized access, confidentiality policies must be applied. Confidentiality of the database is often protected by data encryption or proprietary software. It can be protected by a monitoring system using Markov Chain and Bell-LaPadula Models. In this paper, a two category secured access database model by semi-Markov chains is discussed. This paper simulates this simplified two category secured access database model, and issues on security management are also addressed.

INTRODUCTION

In a typical e-commerce database, data is accessed by the public, twenty four hours per day, and seven days per week (Smith, 2006; Halle & Kikinis, 2004). While the Internet is especially vulnerable to exposing confidential information from e-commerce databases through wire or wireless communications, most of the literature discusses how to protect the data integrity of databases. Confidentiality is often ensured using data encryption, but using encryption often degrades database access speed. Additionally, databases are too frequently used to be securely managed, and data privacy is actually never fully monitored (Becker, Amab, & Serra, 2008). Hence, this presents a major problem for databases in ensuring both confidentiality and performance.

The Bell-LaPadula Model is a confidentiality model for system security and has been used by the United States military for years (Bell & LaPadula, 1973, 1975). Chen, Lee, and Yang (2006) and Chen, Shing, lee, and Shing (2007) use the Bell-LaPadual model to classify suppliers and purchasing companies into different security groups in a supply chain network. Shing, Shing, Chen, and Lee (2006) further suggested modeling the security state transitions by using Markov Chain model in which states were created by use of all the combinations of (Security Clearance, Classification) pairs in the Bell-LaPadula model. Later Shing, Shing, Chen, and Lee (2008, 2009) has extended the model and provided a simulation experiment to prove the effectiveness of their model. This study is to continue Shing and Chen's previous work by providing more explanations of the Markov chain model and to use another simulation set to prove the model's effectiveness.

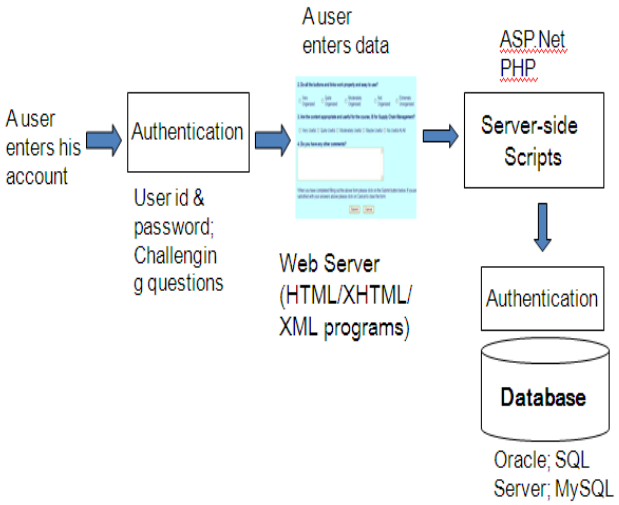
The next section discusses what categories (Security Clearance, Classification) are in the Bell-LaPadula model. Then, it attempts to model a secured database network using a privileged group who may take four different actions in a sequence and a public group who has only two different "read" privileges. Finally, the results from the simulations, as well as several properties, are examined and analyzed.

Secured Database Access in a Web Environment

In a web environment, a user has to use a simple authentication method (i.e. user id and password) to enter his account or the network. When the user enters sensitive data into a database through a web form, the database normally will use another authentication technique to make sure that the user has the right to access the database (See Figure 1). On December 2006, the retail giant TJX company detected a "suspicious software" on its computer system and later it was confirmed an intrusion and data loss. In this case, database break-in was the result of inappropriate data access control and cost the TJX more than 40 millions (Panko, 2009).

There are many methods used to control the confidentiality of data. One of them is the Bell-LaPadula Model which has been used in the military (Bishop, 2003). Assuming in a simplified secured database network, there are only two groups involved: the privileged group and the public group. These privileged groups are called security clearances. In the model, the privileged group can act on two objects. The actions are either to access system files or to use special software tools. For the public group security clearance, users are to read announcements or advertisements on Web pages. However, users in the public group are not allowed to access system files nor to use special software tools. Furthermore, the privileged group members are allowed access to all of the permissions that the public group enjoys. In the next section, we will discuss another method, the semi-Markov chain model, to control the security.

Figure 1: Database Access in a Web Environment.



Semi-Markov Chain Model for Secured Database Access in a Web Environment

Like nature disasters, security threats cannot be completely stopped. However, we can use a probability model (such as Markov chain) to predict the possible security condition of the database systems. A Markov chain is a stochastic process in which the probability of a system state depends only on the previous state (Molloy, 1988). In the following, we use a simple security access model to explain how to use the Markov chain to monitor the security (Bhat, 1972). This example is revised from the Wikipedia (2009). The probabilities of security conditions, given the security on the preceding period, can be represented by a transition matrix:

$$P = \begin{bmatrix} 0.8 & 0.2 \\ 0.5 & 0.5 \end{bmatrix}$$

The matrix P represents the security model in which a secure period is 80% likely to be followed by another secure period, and a non-secure period is 50% likely to be followed by another secure period. The columns can be labeled “secure" and “non-secure" respectively, and the rows can be labeled in the same order. $(P)_{ij}$, the *i*th row and *j*th column of the probability matrix P, is the probability that, if a given period is of type *i*, it will be followed by a period of type *j* (Wikipedia, 2009).

The security on period 0, the initial state, is known to be secure. This is represented by a vector in which the “secure" entry is 100%, and the “non-secure" entry is 0%:

$$X^{(0)} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

The security on period 1 can be predicted by:

$$X^{(1)} = X^{(0)}P = \begin{bmatrix}1 & 0\end{bmatrix} \begin{bmatrix}0.8 & 0.2\\ 0.5 & 0.5\end{bmatrix} = \begin{bmatrix}0.8 & 0.2\end{bmatrix}$$

Thus, there is an 80% chance that period 1 will also be secure. The secure on period 2 can be predicted in the same way:

$$X^{(2)} = X^{(1)}P = X^{(0)}P^2 = \begin{bmatrix}1 & 0\end{bmatrix} \begin{bmatrix}0.8 & 0.2\\ 0.5 & 0.5\end{bmatrix}^2 = \begin{bmatrix}0.74 & 0.26\end{bmatrix}$$

General rules for period n are (Wikipedia, 2009):

$$X^{(n)} = X^{(n-1)}P$$

$$X^{(n)} = X^{(0)}P^n$$

and p_{ij} is the probability of the system in the state *j*, given it was in the state *i*. Furthermore, a semi-Markov chain can be defined as a stochastic process that can have an arbitrary distribution between state changes (Molloy, 1988). A variety of different distributions will be used in the simulation in the following section.

In order to apply semi-Markov chains to the simplified secured database network, all states in the proposed model consist of all the possible combinations of (Security Clearance, Classification) pairs in the Bell-LaPadula model. A detailed calculation of the state transitions can be referred in the paper by Shing et al. (2006). In the next section, we will discuss the simulation experiment and results.

RESEARCH METHOD

In this research, we will use a simulation experiment to explain how to use semi-Markov chain model to monitor the security of the database access in a web environment. The simulation program starts by randomly generating initial states of privileged group's four possible actions and public group's two possible actions. And then we simulate the state transitions based on a randomly generating distribution for each row of the transition probability matrix. In the next section we compare the steady states after simulation runs starting from the same initial states and time=one million.

Table 1 presents a sample run of the semi-Markov chain when both privileged group and other public group's transition probability matrices are independent standard normal distributions. It

also reaches a steady state after time=one million:

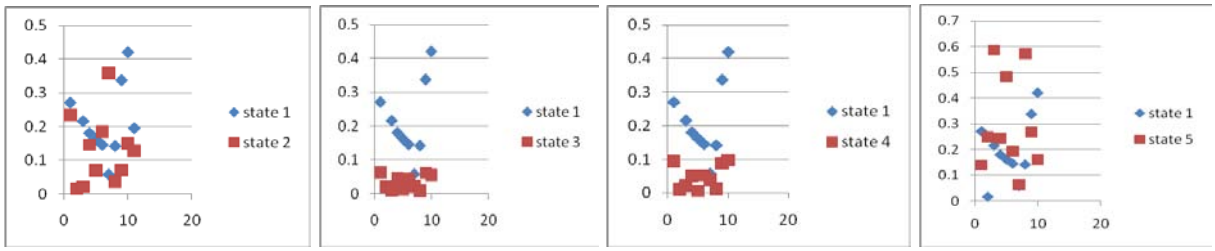
Table 1: A Semi-Markov Chain Simulation Run for 2 category confidential model.

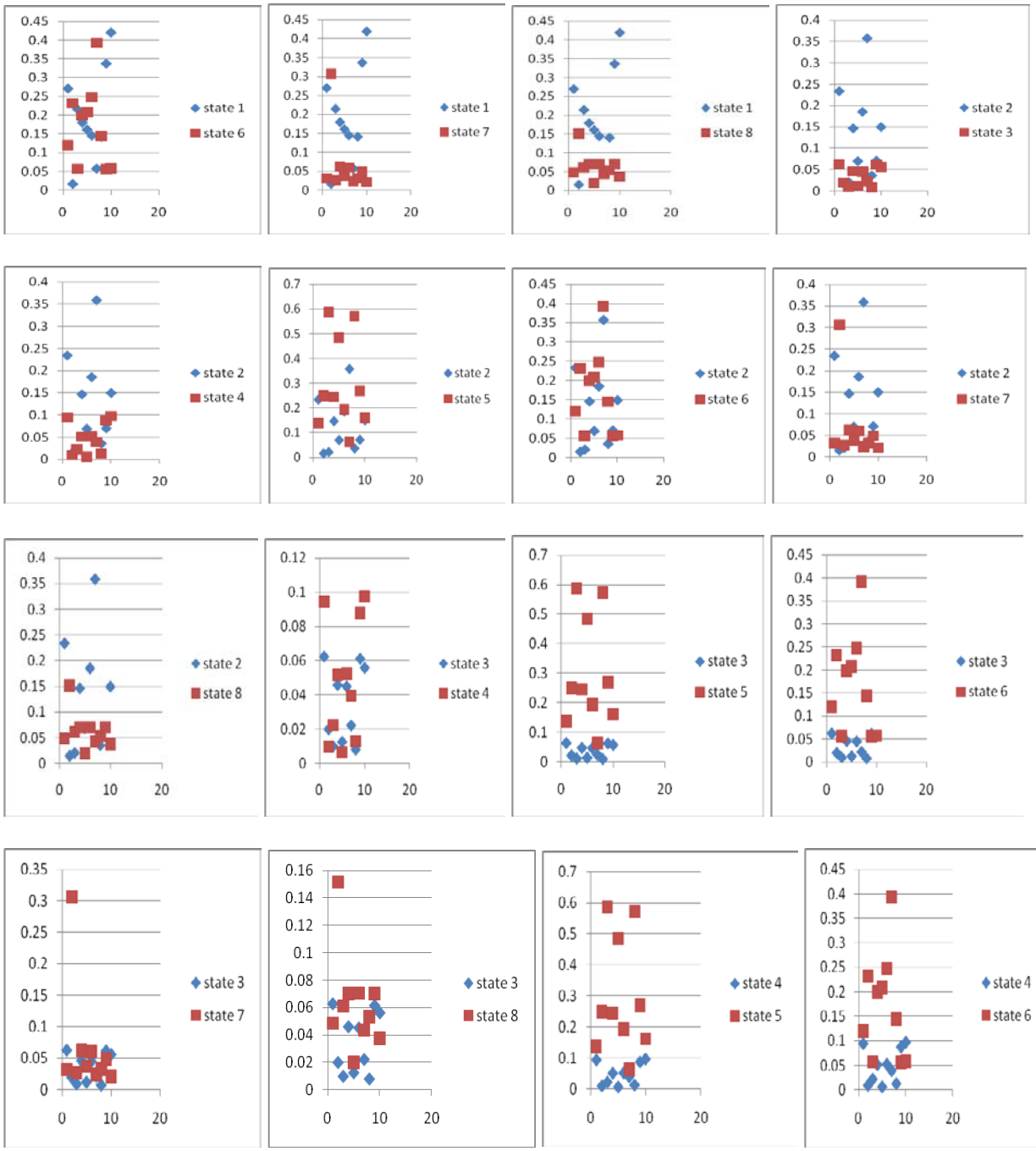
state	1	2	3	4	5	6	7	8
Time=0 state	0.00802	0.15932	0.11954	0.11931	0.01174	0.23284	0.17475	0.17439
State Transition probability matrix T	0.18303	0.36771	0.42340	0.02186	0.23161	0.46547	0.53592	0.02761
	0.26438	0.08511	0.17835	0.41728	0.33452	0.10773	0.22571	0.52810
	0.02827	0.12613	0.00100	0.00718	0.03578	0.15965	0.00136	0.00910
	0.13115	0.02778	0.00399	0.16051	0.16594	0.03511	0.00497	0.20315
	0.11850	0.23829	0.27435	0.01416	0.06992	0.14053	0.16193	0.00830
	0.17122	0.05510	0.11555	0.27030	0.10108	0.03258	0.06819	0.15958
	0.01832	0.08172	0.00064	0.00464	0.01081	0.04820	0.00037	0.00272
	0.08492	0.01796	0.00253	0.10407	0.05012	0.01062	0.00144	0.06134
Time=1000000 state	0.27036	0.234007	0.062476	0.094555	0.138411	0.119802	0.031985	0.048408

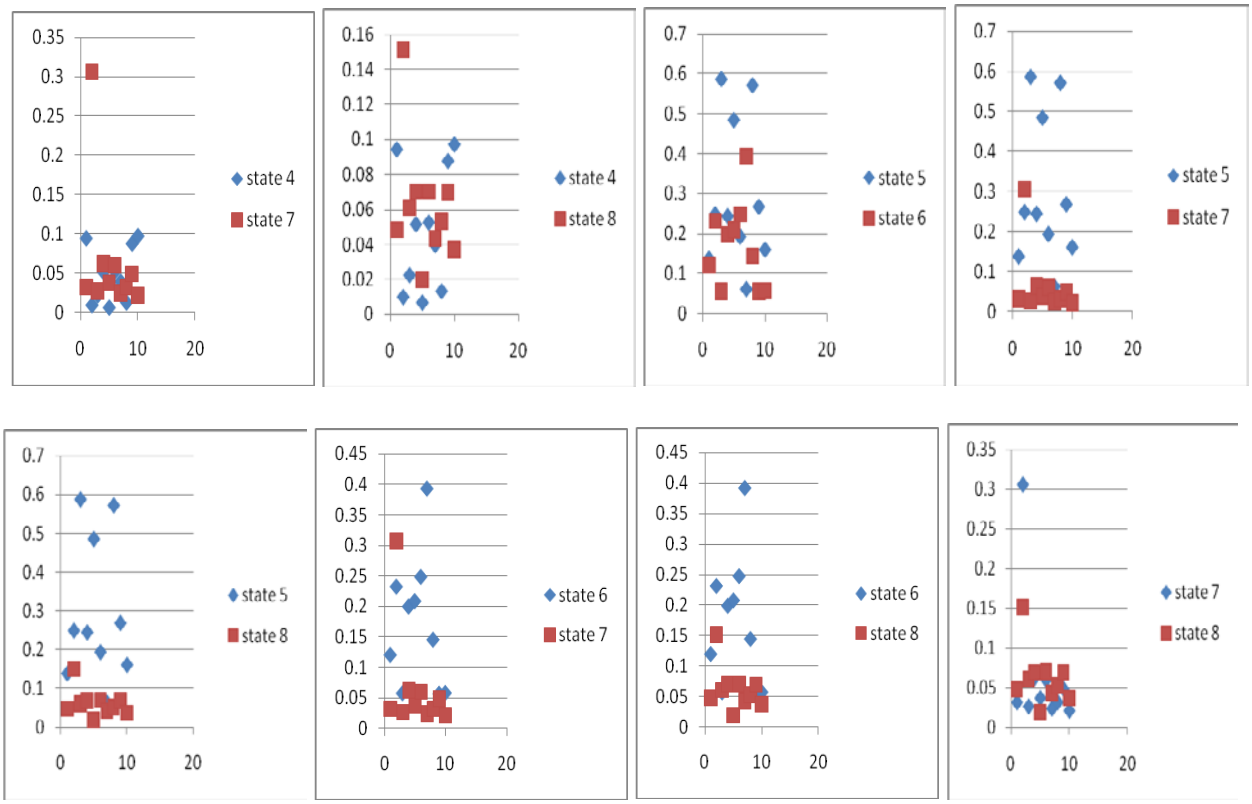
In order to generate the transition probability for a semi-Markov chain, the transition probabilities should be drawn from a truncated probability distribution. For example, there are four states in the privileged group, the transition probability for the last state is generated by subtracting the sum of transition probabilities of the first three states from one. The initial state is set to be $X^{(0)}=(X_1^{(0)}, X_2^{(0)}, \ldots, X_8^{(0)})=(0.008032, 0.159372, 0.119594, 0.119371, 0.011734, 0.232814, 0.174705, 0.174379)$ and $\sum_{i=1}^{8} X_i^{(0)} = 1$. Each simulation runs for one million time units. The simulation runs are randomly repeated for ten times. The mean μ_0 of final states of all 10 runs are created and the variance-covariance matrix Σ is created for each case. Those eight states are correlated which can be shown by drawing scatter plots as in the Figure 2 where the horizontal axis represents the number of runs.

Assume that Σ is known, we can test the hypothesis H_0: $\mu = \mu_0$ vs H_1: $\mu \neq \mu_0$. We reject H_0 at the α level if $n(\bar{y} - \mu_0)' \Sigma^{-1} (\bar{y} - \mu_0) > \chi^2_{\alpha,p}$, where $\bar{y} = \sum_{i=1}^{10} y_i /10$ is the average of n (n=10) final states of all runs and p (p=8)is the total number of states [15]. That is, to reject H_0 at 0.05 level if $10(\bar{y} - \mu_0)' \Sigma^{-1} (\bar{y} - \mu_0) > \chi^2_{0.05,8} = 15.51$ (Note: Σ can be known from analyzing data from a long history or have a substantial evidence to support it).

Figure 2: Scatter plots of eight states when both privileged group and public group are randomly generated uniform (0,1).




state 1
state 6
state 1
state 7
state 1
state 8
state 2
state 3
state 2
state 4
state 2
state 5
state 2
state 6
state 2
state 7
state 2
state 8
state 3
state 4
state 3
state 5
state 3
state 6
state 3
state 7
state 3
state 8
state 4
state 5
state 4
state 6



However, if Σ is unknown, the sample variance-covariance matrix S is used to estimate Σ . Then we can test the hypothesis H_0: μ= μ_0 vs H_1: μ≠ μ_0. We reject H_0 if $10(\bar{y} - \mu_0)' S^{-1} (\bar{y} - \mu_0) > T^2_{\alpha,8,9}$, where $\bar{y} = \sum_{i=1}^{10} y_i /10$ is the average of 10 final states of all runs and T^2 is the Hotelling's T^2 test. And S is the sample variance-covariance matrix, which can be obtained by $\sum_{i=1}^{n} (y_i - \bar{y})(y_i - \bar{y})' / (n-1)$ (Rencher, 1995). S^{-1} is the inverse matrix of S. Table 2 shows mean μ_0 and the sample variance-covariance matrix S of 10 runs of a randomly generated exponential distributions with λ =0.5 for both privileged group and public group transition matrices.

Table 2: Mean μ_0 and the sample variance-covariance matrix S of 10 runs.

	state 1	state 2	state 3	state 4	state 5	state 6	state 7	state 8
run 1	0.035758	0.033229	0.028166	0.063432	0.186915	0.173696	0.147228	0.331576
run 2	0.075491	0.083102	0.063786	0.102328	0.156999	0.172827	0.132656	0.212811
run 3	0.070722	0.074572	0.046805	0.136803	0.144302	0.152158	0.095503	0.279136
run 4	0.167957	0.121822	0.085619	0.236955	0.106324	0.077119	0.054201	0.150003
run 5	0.004344	0.014963	0.013163	0.025601	0.070465	0.242701	0.213505	0.415257
run 6	0.135167	0.128909	0.024098	0.395106	0.062654	0.059753	0.01117	0.183143
run 7	0.029725	0.025859	0.022198	0.044041	0.214277	0.186412	0.160016	0.317473
run 8	0.103477	0.184298	0.177478	0.079137	0.086602	0.154242	0.148535	0.066231

run 9	0.04108	0.027073	0.136061	0.265332	0.046409	0.030585	0.15371	0.29975
run 10	0.034968	0.012358	0.082895	0.228278	0.062572	0.022113	0.148333	0.408484
Average	0.0698689	0.0706185	0.0680269	0.1577013	0.1137519	0.1271606	0.1264857	0.2663864
Covariance	0.0054617	0.0052972	0.002789	0.0062042	0	0	0	0
	0.0052972	0.0061768	0.0040692	0.0043542	0	0	0	0
	0.002789	0.0040692	0.0056912	0.0030693	0	0	0.0015227	0
	0.0062042	0.0043542	0.0030693	0.0125776	0	0	0	0
	0	0	0	0	0.0061557	0.0052186	0.0020248	0.0017799
	0	0	0	0	0.0052186	0.0078473	0.0049162	0.0033331
	0	0	0.0015227	0	0.0020248	0.0049162	0.0061214	0.0063737
	0	0	0	0	0.0017799	0.0033331	0.0063737	0.011858

To test the hypothesis H_0: $\mu = \mu_0$ vs H_1: $\mu \neq \mu_0$, where μ_0 =

0.0698689 0.070618 0.068026 0.157701 0.113751 0.127160 0.126485 0.266386

The average of ten observations is $\bar{y}$ =

0.0558786 0.0584626 0.0638141 0.1348081 0.130934 0.118523 0.1117931 0.3257857

Table 2: Average μ_0 and the sample (variance-) covariance matrix S of 10 runs of a randomly generated exponential distribution with λ =0.5

From Table 2, $10(\bar{y} - \mu_0)' S^{-1} (\bar{y} - \mu_0) = 67326 > T^2_{0.05,8,9} = 697.356$. The null hypothesis is rejected at 0.05 level. Therefore the security may have been breached.

CONCLUSION

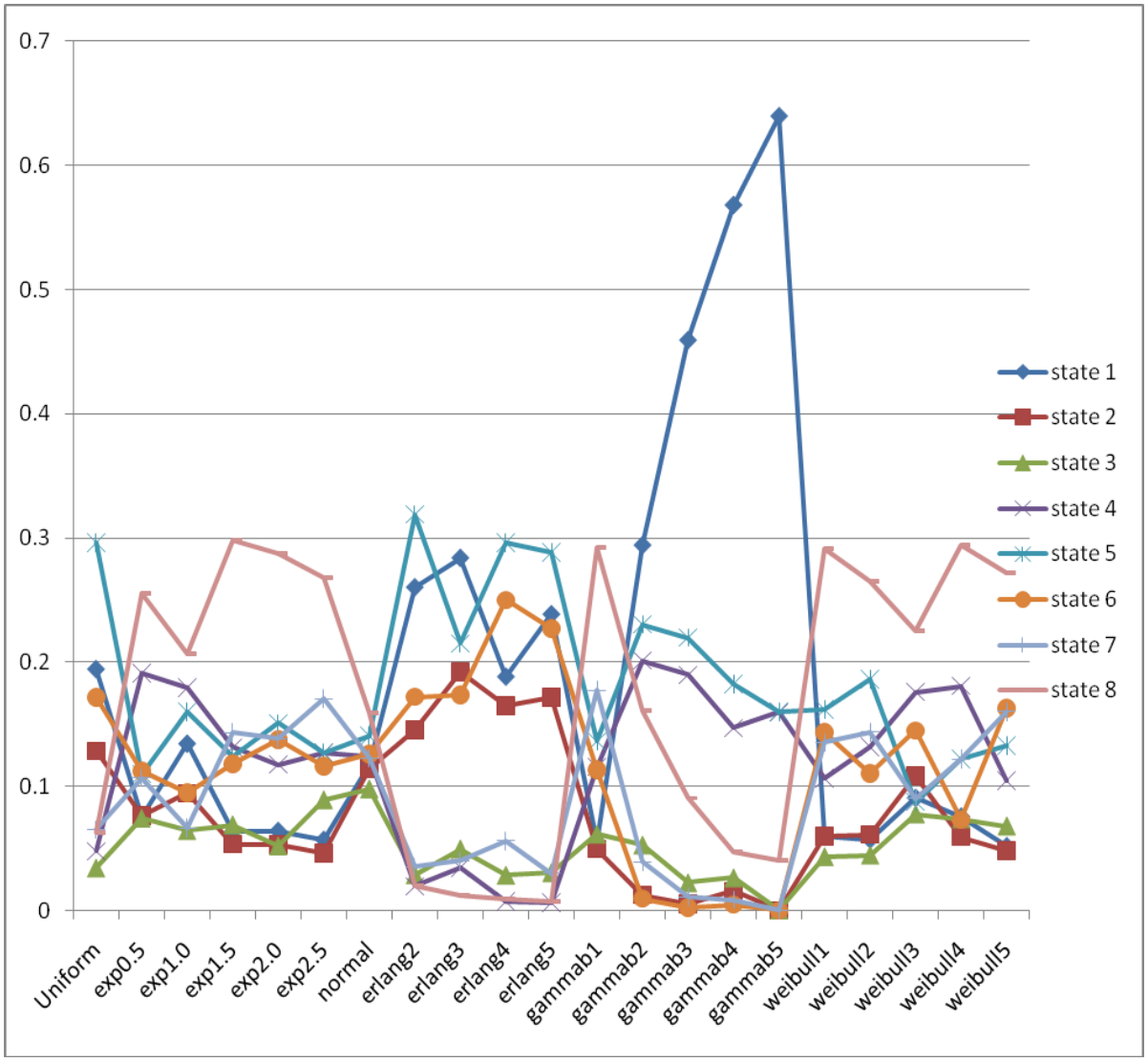
There are six different truncated distributions generated for privileged and public transition probability matrices. They are Erlang of order k (k=2, 3, 4, 5), exponential with parameter λ (λ=0.5, 1, 1.5, 2, 2.5), gamma with parameter θ=1 and β (β=1, 2, 3, 4, 5), standard normal, uniform (0,1) and Weibull distributions with ν=0, α=0.5, and β (β=1, 2, 3, 4, 5). To generate the transition probability matrix for each distribution, seven random variates were generated first. Then, they are sorted, and the probabilities up to the random variates were calculated according to trapezoidal rule. The probability for the last state was found by subtracting the sum of those probabilities from one. The algorithms for generating random variates for those distributions are given in (Banks, Carson, & Nelson, 1996). Because every entry of both privileged and public transition matrices is non-zero, they all reach equilibrium states. Table 3 shows the average of final states after 10 runs for those randomly generated distributions.

Table 3: Average of final state after 10 runs.

	Average of 10 Runs							
	state 1	state 2	state 3	state 4	state 5	state 6	state 7	state 8
Uniform	0.19422	0.12855	0.03426	0.04751	0.29617	0.17146	0.06513	0.06258
exp0.5	0.07497	0.07591	0.07414	0.19103	0.10965	0.11217	0.10686	0.25518
exp1.0	0.13429	0.09438	0.06419	0.17969	0.15971	0.09497	0.06636	0.20652
exp1.5	0.06333	0.05342	0.06896	0.13135	0.12374	0.11822	0.14313	0.29786
exp2.0	0.06396	0.05304	0.05194	0.11742	0.1508	0.13737	0.13803	0.28746
exp2.5	0.05682	0.04623	0.08881	0.12678	0.12676	0.11605	0.17069	0.26776
normal	0.11678	0.11409	0.09745	0.12391	0.14061	0.12564	0.12248	0.15896
erlang2	0.26023	0.14558	0.02886	0.01957	0.31886	0.17209	0.03508	0.01979
erlang3	0.28381	0.19213	0.04952	0.03424	0.21496	0.17364	0.03996	0.01179
erlang4	0.18822	0.16456	0.02851	0.00741	0.29623	0.24997	0.05605	0.00909
erlang5	0.23856	0.17182	0.03057	0.00643	0.28817	0.22721	0.03008	0.00718
gammab1	0.05515	0.04942	0.06129	0.11553	0.13612	0.11308	0.17743	0.29209
gammab2	0.29418	0.01257	0.05268	0.20122	0.23032	0.00947	0.03895	0.16066
gammab3	0.45924	0.00526	0.02247	0.18966	0.21954	0.00255	0.01089	0.09038
gammab4	0.56793	0.01547	0.02639	0.14695	0.18249	0.00479	0.00865	0.04735
gammab5	0.63954	0.00015	0.00059	0.15948	0.16018	2.65E-5	0.00015	0.03996
weibull1	0.05932	0.05962	0.04338	0.10617	0.16184	0.14339	0.13527	0.29094
weibull2	0.05702	0.06108	0.04454	0.13189	0.18618	0.11057	0.14359	0.26524
weibull3	0.09064	0.10824	0.07744	0.17582	0.08737	0.14493	0.09026	0.22521
weibull4	0.07561	0.05918	0.07356	0.18066	0.12173	0.07298	0.12198	0.29421
weibull5	0.05169	0.04821	0.06791	0.10444	0.13267	0.16289	0.15999	0.27215

To visualize the comparison of the averaged steady states in Table 2, Figure 3 shows only state 3 in the steady states is about the same for all different distributions. The rest states can be a lot different.

Figure 3: Comparison of steady states for different distributions.



By constantly checking the state transition matrix, we can find out whether the system will reach a steady state. Any state that does not belong to one of the possible eight states violates the security requirement. Besides, if any state probability p(s) at time s from the simulation run shows any abnormality compared to data collected, then it may have a security breach. For example, if a public group accesses system files, then the system will warn the security privileged group about security leak and suggest the group to take actions. Overall, the proposed model can help managers to understand the confidential status of each public group and then implement necessary security strategies.

REFERENCES

Banks, J., Carson J., & Nelson, B. (1996). *Discrete Event System Simulation*, Upper Saddle River, NJ: Prentice Hall.

Becker, A., Amab, A., & Serra, M. (2008). Assessing privacy criteria for DRM using EU privacy legislation, *Proceedings of the 8th ACM Workshop on Digital Rights Management.*

Bell, D., & LaPadula, L. (1973). Secure computer systems: Mathematical foundations, *Technical Report MTR-2574*, I, Bedford, MA: MITRE Corporation.

Bell, D., & LaPadula, L. (1975). Secure computer system: Unified exposition and multics Interpretation, *Technical Report MTR-2997*, Rev. 1, Bedford, MA: MITRE Corporation.

Bhat, N. (1972). *Elements of Applied Stochastic Processes*, New York, NY: John Wiley & Sons.

Bishop, M. (2003). *Computer Security*, Boston, MA: Addison-Wesley.

Chen, K. L., Lee, H., & Yang, J. (2006). Security considerations on the design of supply chain networks, *Proceedings of the Southwest Division of the Decision Sciences Institute (SWDSI)*, 14(3*).*

Chen, K. L., Shing, M., Lee, H. & Shing, C. (2007). Modeling in confidentiality and integrity for a supply chain network, *Communications of IIMA*, 7(1), 41-48.

Halle, M., & Kikinis, R.(2004). Flexible frameworks for medical multimedia, *Proceedings of the 12th Annual ACM International Conference on Multimedia* .

Molloy, M. (1988). *Fundamentals of Performance Modeling*, Macmillan Publishing.

Panko, R. (2009). *Corporate Computer and Network Security* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

Rencher, A. (1995). *Methods of Multivariate Analysis.* New York: John Wiley & Sons.

Shing, M., Shing, C., Chen, K., & Lee, H. (2006). Security Modeling on the supply chain networks, *Journal of Systemics, Cybernetics and Informatics*, 5(5), 53-58.

Shing, M., Shing, C., Chen, K., & Lee, H. (2008). A Simulation Study of Confidentiality Modeling in a Secured Supply Chain Network, *Proceedings of International Symposium on Intelligent. Information Technology Application conference, Dec. 22-23, 2008.*

Shing, M., Shing, C., Chen, K., & Lee, H. (2009). Confidentiality modeling and simulation and validation in a simplified database access, *International Journal of Computational Biology and Drug Design*, 2(3), 252-263.

Smith, *A.* (2006). Supply chain management using electronic reverse auctions: a multi-firm case study, *International Journal of Services and Standards*, 2(2), 176 – 189.

Wikipedia (2009). Examples of Markov Chains. Retrieved from http://en.wikipedia.org/wiki/Examples_of_Markov_chains, Oct 13, 2009.