AN IMPROVED METHODOLOGY FOR SYSTEM THREAT ANALYSIS USING MULTIPLE-VALUED LOGIC AND CONDITIONAL PROBABILITIES

Theodore W. Manikas, Mitchell A. Thornton
Darwin Deason Institute for Cyber Security, Dept. of Computer Science and Engineering, Southern Methodist University, Dallas, Texas, USA
(manikas, mitch@lyle.smu.edu)
Shinobu Nagayama
Dept. of Computer and Network Engineering, Hiroshima City University, Hiroshima, JAPAN
(s_naga@hiroshima-cu.ac.jp)

ABSTRACT

The operation of large systems is affected by both natural and intentional threats. Many modern system analysis methods use binary logic models to represent system operations. However, these models are limited as they can only represent two operational states: full operation mode or complete failure. In order to represent intermediate system states, such as partial failures, multiple-valued logic models (MDDs) must be used. Because threats that result in system degradation often have a conditional nature, these conditional probabilities must be computed in an efficient manner. An improved technique based upon the notion of conditional probability table cell indices is described and shown to allow for efficient scalability.

INTRODUCTION

An important part of system analysis is the determination of risks for system threats. Common risks include system vulnerabilities that can be exploited by an external attacker, such as Stuxnet (Karnouskos, 2011) and Heartbleed (Tsoutsos & Maniatakos, 2014). Various tree-like data structures have been developed to represent possible system threats, such as fault trees (Vesely, Goldberg, Roberts, & Haasi, 1981) or attack trees (Schneier, 1999).

A common structure for fault representation is the binary decision diagram (BDD), which is a rooted directed acyclic graph (DAG) that can be used to represent large switching functions in an efficient manner (Bryant, 1986). Figure 1 shows a BDD for the function *f(A, B)* = *A* and *B*. In this diagram, if both *A* and *B* have a logic value of 1, then the output value for *f(A, B)* is 1. Otherwise, the output value is 0.

The BDD structure has been applied to many areas including the representation of fault trees (Rauzy, Gauthier, & Leduc, 2007)(Xing & Dai, 2009)(Yevkin, 2009)(Mahdi & Nadji, 2013). Furthermore, efficient software is readily available to manipulate BDDs and a variety of heuristics and strategies have been adapted for use with fault trees (Rauzy et al., 2007)(Yao Cai, Zhengjiang Liu, & Zhaolin Wu, 2009). However, these structures are based on a binary model whereby a system either operates in a fully functional or a complete failure mode. Modeling different operational modes other than the binary case of failure or normal operation are critical in analyzing large systems in the presence of threats.

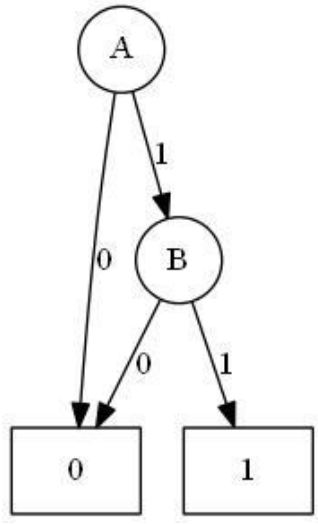

Fig. 1 BDD for function *f* = *A* and *B*

To address this limitation we need to expand our models to handle more than 2 logic states. For example, we may want to represent the following states of a system: (2) fully operational, (1) partially operational, and (0) non-operational. This system has 3 states, so it is represented by a ternary system (radix 3). In general, systems with radix > 2 are called multiple-valued logic (MVL) systems. For example, we can expand the binary AND function into a radix-3 MIN function as shown in Table 1.

In the case of MVL, an extension to the BDD construct has been developed and implemented called the Multiple-Valued Decision Diagram (MDD) (Miller & Thornton, 2007). Similar to the BDD, the MDD is also a DAG and it contains a maximum of *p* terminal nodes, where each terminal node is labeled by a distinct logic value in the range [0, *p*-1]. Figure 2 shows an example of an MDD for the radix-3 function *f* = MIN(*A, B*) of Table 1.

Table 1 Truth table for 2-input MIN function

A	B	f(A,B) = MIN(A,B)
0	0	0
0	1	0
0	2	0
1	0	0
1	1	1
1	2	1
2	0	0
2	1	1
2	2	2

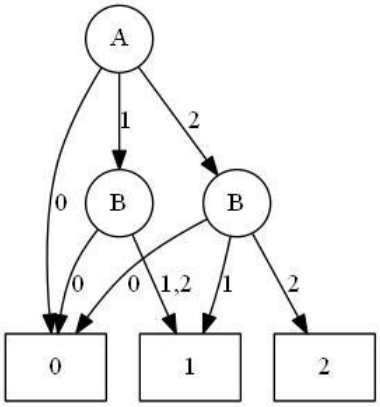

Fig. 2 MDD for radix-3 MIN function

Most practical systems have many interdependent components, so the conditional probabilities of input states must be considered during system analysis. While conditional probability analysis for binary systems has been widely studied, there has been limited research in this area for multiple-valued logic systems with radices > 2. Preliminary work in this area has been done by (Manikas, Feinstein, & Thornton, 2012)(Nagayama, Sasao, Butler, Thornton, & Manikas, 2014)(Thornton, Manikas, Szygenda, & Nagayama, 2014)(Yuchang Mo, Liudong Xing, & Amari, 2014) for application to multiple-valued logic decision diagrams for system analysis. These methods typically represent system operation using MDD's and apply graph traversal methods to calculate the probabilities of system output states. While these approaches are effective for handling small systems, they can become unwieldy as system size grows.

This paper describes an alternate approach to calculating system output state probabilities, given the conditional probabilities of input states for a given system. The approach described in this paper expands on the traditional conditional probability theory (which focuses on binary systems) to multiple-valued logic systems.

CONDITIONAL PROBABILITIES FOR BINARY SYSTEMS

The traditional approach to calculating conditional probabilities assumes a binary logic system: each event is either true or false. For example, we may have two events $x_1(j)$ and $x_0(i)$, where $P(x_1(j))$ is the probability that $x_1(j)$ has logic value j (0 or 1), and $P(x_0(i))$ is the probability that $x_0(i)$ has logic value i (0 or 1). From general probability theory (Douglas Montgomery & Runger, 2003), the conditional probability of event $x_0(i)$ given the occurrence of event $x_1(j)$ is $P(x_0(i)|x_1(j))$ as shown in Eq. 1.

$$P(x_0(i) \mid x_1(j)) = \frac{P(x_1(j) \cap x_0(i))}{P(x_1(j))} \quad (1)$$

Table 2 Conditional probabilities for a radix-2, 2-input system

	$x_0=0$	$x_0=1$	total
$x_1=0$	$P(x_1(0) \cap x_0(0))$	$P(x_1(0) \cap x_0(1))$	$P(x_1(0))$
$x_1=1$	$P(x_1(1) \cap x_0(0))$	$P(x_1(1) \cap x_0(1))$	$P(x_1(1))$
total	$P(x_0(0))$	$P(x_0(1))$	1

Table 3 Cell table indices for a radix-2, 2-input system

	$x_0=0$	$x_0=1$	total
$x_1=0$	0	1	$P(x_1(0))$
$x_1=1$	2	3	$P(x_1(1))$
total	$P(x_0(0))$	$P(x_0(1))$	1

Table 4 Conditional probabilities for 2-input binary system

	$x_0=0$	$x_0=1$	total
$x_1=0$	0.2	0.4	0.6
$x_1=1$	0.1	0.3	0.4
total	0.3	0.7	1

The conditional probabilities are determined as shown in Table 2, where:

- $P(x_1 \cap x_0)$ = probability that both x_1 and x_0 occur.
- $P(x_1)$ = total probability that x_1 occurs = $P(x_1 \cap x_0')$ + $P(x_1 \cap x_0)$.

The index i of each table cell where the values intersection is found by $i = 2x_1 + x_0$. For example, the table cell for $P(x_1(1) \cap x_0(0))$ is $i = 2(1) + 0 = 2$. The corresponding table cell indices are shown in Table 3.

Now assume we have a 2-input binary system that implements the AND function and has the conditional probabilities shown in Table 4. Using the approach of (Manikas, Feinstein, & Thornton, 2012), we can determine the probabilities of obtaining output value 0 (Eq. 2) and output value 1 (Eq. 3).

$$\begin{aligned} P(0) &= P(x_1(0)) + P(x_1(1))P(x_0(0) \mid x_1(1)) \\ &= 0.6 + (0.4)(0.25) = 0.7 \end{aligned} \quad (2)$$

$$\begin{aligned} P(1) &= P(x_1(1))P(x_0(1) \mid x_1(1)) \\ &= (0.4)(0.75) = 0.3 \end{aligned} \quad (3)$$

However, we can also calculate these output probabilities using the values of the cell node probabilities. Note that for the AND function, cell nodes 0, 1, and 2 map to output value 0, while cell node 1 maps to output value 1. Therefore, the output probabilities can be calculated by summing the probability values of the cell nodes that map to the particular output node. For the AND function, this would produce the output probabilities as shown in Eq. 4 and Eq. 5.

$$P(0) = \sum cellNodes(0,1,2) \quad (4)$$
$$= 0.2 + 0.4 + 0.1 = 0.7$$

$$P(1) = \sum cellNodes(3) = 0.3 \quad (5)$$

Note that this approach gives the same output probability results as the method of (Manikas et al., 2012), but with simpler calculations. Therefore, this approach is more computationally efficient.

CONDITIONAL PROBABILITIES FOR TERNARY SYSTEMS

We can further expand our system to a radix-3 (ternary), 3-input system. The corresponding table cell indices are shown in Table 5 for x_2=0, Table 6 for x_2=1, and Table 7 for x_2=2.

Now, assume that we have a SCADA (Supervisory Control And Data Acquisition) system (Rautmare, 2011) that has three components: a nuclear reactor, a wastewater treatment system, and oil refinery. Also assume that these systems have three possible levels of operation: fully operational, partially operational, or non-operational. We can model this system as a radix-3, 3-input system. The inputs are the operation states of the nuclear reactor (x_2), the wastewater treatment system (x_1), and the oil refinery (x_0). The operation levels are (2) fully operational, (1) partially operational, and (0) non-operational.

Table 5 Cell table indices for a radix-3, 3-input system: $x_2 = 0$

	x_0=0	x_0=1	x_0=2	total
x_1=0	0	1	2	P(x_1(0))
x_1=1	3	4	5	P(x_1(1))
x_1=2	6	7	8	P(x_1(2))
total	P(x_0(0))	P(x_0(1))	P(x_0(2))	1

Table 6 Cell table indices for a radix-3, 3-input system: $x_2 = 1$

	x_0=0	x_0=1	x_0=2	total
x_1=0	9	10	11	P(x_1(0))
x_1=1	12	13	14	P(x_1(1))
x_1=2	15	16	17	P(x_1(2))
total	P(x_0(0))	P(x_0(1))	P(x_0(2))	1

Table 7 Cell table indices for a radix-3, 3-input system: $x_2 = 2$

	x_0=0	x_0=1	x_0=2	total
x_1=0	18	19	20	P(x_1(0))
x_1=1	21	22	23	P(x_1(1))
x_1=2	24	25	26	P(x_1(2))
total	P(x_0(0))	P(x_0(1))	P(x_0(2))	1

Table 8 Conditional probabilities for SCADA system inputs: $x_2 = 0$

	x_0=0	x_0=1	x_0=2
x_1=0	0.02	0.055	0.01
x_1=1	0.04	0.02	0.0075
x_1=2	0.025	0.05	0.02

Table 9 Conditional probabilities for SCADA system inputs: $x_2 = 1$

	x_0=0	x_0=1	x_0=2
x_1=0	0.0215	0.04	0.05
x_1=1	0.04	0.06	0.001
x_1=2	0.025	0.075	0.0475

Table 10 Conditional probabilities for SCADA system inputs: $x_2 = 2$

	x_0=0	x_0=1	x_0=2
x_1=0	0.04	0.025	0.075
x_1=1	0.02	0.0375	0.05
x_1=2	0.06	0.045	0.04

In addition, we will assume that we know the conditional probabilities for the inputs of this SCADA system for a given system threat, such as Stuxnet. The conditional probabilities indicate the relationship between operating states for the SCADA system components during an attack. Table 8 shows the conditional probabilities for x_2=0, while Table 9 shows these values for x_2=1 and Table 10 shows the values for x_2=2. The operation of the SCADA system is shown in the MDD of Fig. 3 and in the truth table of Table 11. Note that the probabilities for each row of Table 11 correspond to the probabilities of each cell in Tables 8, 9, and 10.

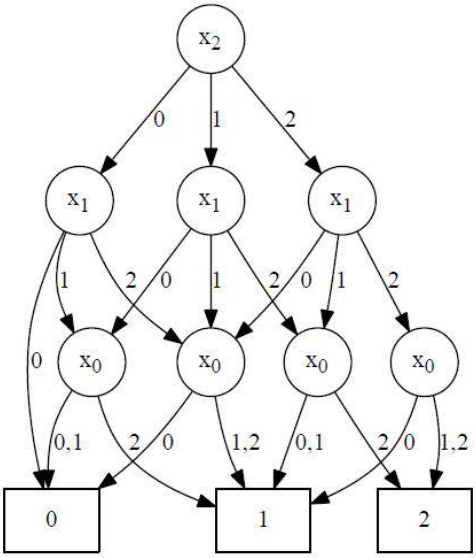

Figure 3 MDD for the SCADA system operation

Table 11 Truth table for SCADA system operation

row	x2	x1	x0	output F	prob.
0	0	0	0	0	0.02
1	0	0	1	0	0.055
2	0	0	2	0	0.01
3	0	1	0	0	0.04
4	0	1	1	0	0.02
5	0	1	2	1	0.0075
6	0	2	0	0	0.025
7	0	2	1	1	0.05
8	0	2	2	1	0.02
9	1	0	0	0	0.0215
10	1	0	1	0	0.04
11	1	0	2	1	0.05
12	1	1	0	0	0.04
13	1	1	1	1	0.06
14	1	1	2	1	0.001
15	1	2	0	1	0.025
16	1	2	1	1	0.075
17	1	2	2	2	0.0475
18	2	0	0	0	0.04
19	2	0	1	1	0.025
20	2	0	2	1	0.075
21	2	1	0	1	0.02
22	2	1	1	1	0.0375
23	2	1	2	2	0.05
24	2	2	0	1	0.06
25	2	2	1	2	0.045
26	2	2	2	2	0.04

We can then determine the output probabilities of the SCADA system by summing the cell node probability values that correspond to each output value of F. First, we calculate P(0) by summing the probabilities for the rows where $F = 0$ (Eq. 6). Similarly, P(1) is calculated by summing the probabilities for the rows where $F = 1$ (Eq. 7), and P(2) is calculated by summing the probabilities for the rows where $F=2$. The results indicate that for our given attack scenario, the SCADA system is most likely to operate in a partially operational or degraded state (logic level 1).

$$P(0) = \sum rows(0,1,2,3,4,6,9,10,12,18) = 0.3115 \quad (6)$$

$$P(1) = \sum rows(5,7,8,11,13,14,15,16,19,20,21,22,24) = 0.506 \quad (7)$$

$$P(2) = \sum rows(17,23,25,26) = 0.1825 \quad (8)$$

CONCLUSION

For systems whose threat scenarios can be modeled by multiple-valued logic and conditional probabilities, we have developed an analysis approach to determine the probability of system outcomes for the given threat. Previous methods required excessive path traversals to compute conditional probability values. The new method described here is based on the notion of a conditional table cell and its location. Thus, the resulting computations when implemented over MDDs results in an improved method that requires less computation and allows the technique to become more scalable for large systems.

REFERENCES

Bryant, R. E. (1986). Graph-Based Algorithms for Boolean Function Manipulation. *Computers, IEEE Transactions on*, *C-35*(8), 677 –691.

Karnouskos, S. (2011). Stuxnet worm impact on industrial cyber-physical system security. *IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society*, 4490–4494.

Mahdi, I., & Nadji, B. (2013). Application of the binary decision diagram (BDD) in the analysis of the reliability of the inverters. *Power Engineering, Energy and Electrical Drives (POWERENG), 2013 Fourth International Conference on*, 1265–1271.

Manikas, T. W., Feinstein, D. Y., & Thornton, M. A. (2012). Modeling Medical System Threats with Conditional Probabilities Using Multiple-Valued Logic Decision Diagrams. In *Multiple-Valued Logic (ISMVL), 2012 42nd IEEE International Symposium on* (pp. 244–249).

Miller, D. M., & Thornton, M. A. (2007). *Multiple Valued Logic: Concepts and Representations*. Morgan & Claypool Publishers.

Douglas Montgomery, & Runger, G. (2003). *Applied Statistics and Probability for Engineers* (3rd Ed.). John Wiley & Sons.

Nagayama, S., Sasao, T., Butler, J. T., Thornton, M. A., & Manikas, T. W. (2014). Analysis Methods of Multi-state Systems Partially Having Dependent Components Using Multiple-Valued Decision Diagrams. *Multiple-Valued Logic (ISMVL), 2014 IEEE 44th International Symposium on*, 190–195.

Rautmare, S. (2011). SCADA system security: Challenges and recommendations. *India Conference (INDICON), 2011 Annual IEEE*, 1–4.

Rauzy, A. B., Gauthier, J., & Leduc, X. (2007). Assessment of large automatically generated fault trees by means of binary decision diagrams. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, *221*(2), 95 –105.

Schneier, B. (1999). Attack trees. *Dr. Dobb's Journal*, *24*(12), 21 – 21.

Thornton, M. A., Manikas, T. W., Szygenda, S. A., & Nagayama, S. (2014). System Probability Distribution Modeling Using MDDs. *Multiple-Valued Logic (ISMVL), 2014 IEEE 44th International Symposium on*, 196–201.

Tsoutsos, N. G., & Maniatakos, M. (2014). Trust No One: Thwarting "Heartbleed" Attacks Using Privacy-Preserving Computation. *VLSI (ISVLSI), 2014 IEEE Computer Society Annual Symposium on*, 59–64.

Vesely, W. E., Goldberg, F. F., Roberts, N. H., & Haasi, D. F. (1981). *Fault tree handbook* (No. NUREG-0492). U.S. Nuclear Regulatory Commission.

Xing, L., & Dai, Y. (2009). A New Decision-Diagram-Based Method for Efficient Analysis on Multistate Systems. *Dependable and Secure Computing, IEEE Transactions on*, *6*(3), 161 –174.

Yao Cai, Zhengjiang Liu, & Zhaolin Wu. (2009). Improvement of Fault Tree Analysis in Formal Safety Assessment Using Binary Decision Diagram. *Information Science and Engineering (ICISE), 2009 1st International Conference on*, 4330–4333.

Yevkin, O. (2009). Truncation approach with the decomposition method for system reliability analysis. In *2009 - Annual Reliability and Maintainability Symposium, RAMS 2009, January 26, 2009 - January 29, 2009* (pp. 430–435). Fort Worth, TX, United states: Institute of Electrical and Electronics Engineers Inc.

Yuchang Mo, Liudong Xing, & Amari, S. V. (2014). A Multiple-Valued Decision Diagram Based Method for Efficient Reliability Analysis of Non-Repairable Phased-Mission Systems. *Reliability, IEEE Transactions on*, *63*(1), 320–330.