A Novel Inference-based Approach to Evaluate Failure Interdependency in Access Control Models

Saad Saleh Alaboodi, Gordon B. Agnew
Electrical and Computer Engineering
University of Waterloo, Waterloo, ON, Canada

Abstract

*Demands for better design and analysis of access controls require system-level evaluation models that can facilitate a quantitative and consistent study of operational capabilities and economics of access control implementations. Previous works on access control models are mainly centered on the access interaction between system subjects and objects with respect to rights, addressing their basic security goals, thus failing to address other dependability attributes. To address this shortcoming, we first propose the abstraction of a computing system into: objects and rights of subjects (called in this paper **assets** and **controls**, respectively) to study the unavoidable failure interdependency between these two classes, a perspective that can be a basis for various failure-related assessment methods. We then propose a modeling technique that probabilistically captures the interaction between assets and controls into a graph theoretic paradigm; we specifically show how Bayesian Networks (BNs) can model this dilemma. This paper presents the proposed abstraction, modeling formalism, and associated notation, along with a demonstration example of various useful inferences and further research directions.*

Keywords: *Access control, failure interdependency, Bayesian Networks, asset-control graph, asset-control modeling, security engineering, security economics.*

1. Introduction

Access controls are indispensable mechanisms for protecting access to resources of computing and communication systems. The main purpose of access controls in principle is to limit the activities of legitimate subjects [1]. Access control models can generally be classified into three different categories or policies: Role-based Access Control (RBAC), Discretionary Access Control (DAC), and Mandatory Access Control (MAC). In an RBAC, access permissions are granted to roles, corresponding to specific job functions or activities [2], [3]. DAC refers to models where access permissions are determined by the owner of an object; and MAC refers to models where access permissions are allowed if and only if rules exist that allow a given user access to a resource [1], [3]. These models are mainly designed to protect access of subjects to objects in accordance with the three main security goals: confidentiality, integrity, and availability.

Clearly, these access control models are based on access interactions between a system's subjects and objects with respect to its allowed rights or permissions. Their implementations can be seen as a set of different, interacting access mechanisms, deployed to collectively prevent any deviation from the intended access of subjects to objects. Thus, the reliability of access controls for physical and computer security is vital for protecting the resources of computing and communication systems. So, access control systems must first be secure in order to be dependable for controlling access to a system's resources.

However, the analysis, design, and implementation of access control models are faced with many challenges nowadays. First, many aspects related to access control functions do not enjoy full independence while in operation: (1) access control-related services (e.g., access authentication, auditing, and administration services); (2) goals sought by access controls (i.e., confidentiality, integrity, and availability); (3) services provided (i.e., physical security and computer security); and (4) security processes implemented (i.e., prevention, detection, and recovery). That is, in a way or another, these aspects interact across different platforms, programs, processes, or users, leading to build a certain interdependency while in operation.

Second, access controls may fail separately or jointly in various forms, either due to malicious causes or nonmalicious causes. For example, an authorization module (perhaps as an access control) on a particular database platform (as an asset) may fail due to design flaws (as a nonmalicious failure) or brute force attacks (as a malicious failure), with or without corrupting the database itself. Such failures also may extend to affect other assets due to the interdependencies involved with those assets.

Third, the design and implementation of reliable access controls are getting more problematic as ubiquitous computing and Cloud-based applications are getting more popular. This transition has led to create unprecedented challenges to defend the

principles of information security and privacy nowadays.

In light of the above challenges, the result is a set of interconnected, interdependent system components, or subsystems, on which the system's owner and users depend, jointly defining the operational security level of access control models. Therefore, the identification and evaluation of this interdependency is crucial to meeting the security goals of access control models, creating the need for better system-level evaluation methods of access control implementations.

In particular, there is a need to identify such a complex interdependency behaviour between assets and controls in order to properly model access controls for greater security and operational performance. This need requires models that facilitate consistent, combined study of operational capability and economics of access control implementations. This combination is necessary to properly align incentives to protecting a system with the "suffer" from its failure [4].

However, regardless of the advancement and associated complexity in computing and communication technologies, systems can still be abstracted into two main classes: assets, which are entities that perform system's primary tasks; and access controls, which are entities that perform the required access protection of such assets. Assets can be low-level objects such as data files or high-level ones such as databases and applications. Similarly, access controls can be low-level, integrated mechanisms such as authorization modules or high-level, standalone ones such as firewalls. The failure of access controls leads to exposing system assets according to their dependency with, and the coverage of, such controls, which could lead to catastrophic system damages.

The above discussion leads us to argue that the security element in access control architectures is twofold, as shown in Fig. 1. The first aspect (Fig. 1a)

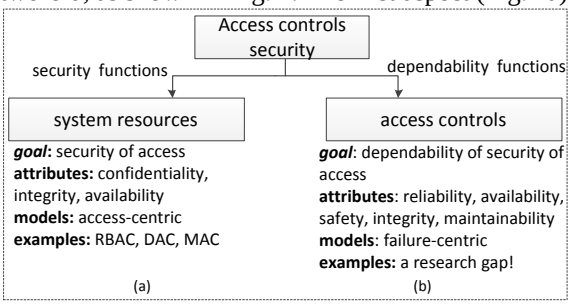

Figure 1. Security aspects in access control architectures. (a) security of access to system resources. (b) operational capabilities and resilience to failure of access controls themselves.

is the security of access to system resources, as implemented by the access control model in place. The second aspect (Fig. 1b) is the operational capability and resilience to failure of access controls. The former aspect is more about the security functions provided by access controls, thus it is centered on access interactions between subjects and objects with respect to rights, as realised by the current studies on access control models. The latter aspect is more about the dependability and reliability functions of access controls themselves, thus it should be centered on failure dynamics between assets and controls, as proposed in this work. We use the following example to further clarify this argument. The cryptanalyst will better design and analyze the strengths of crypto primitives implemented over a particular access control. But, the reliability analyst will better design and analyze the dependability features of that access control, considering its failure behaviour and associated interdependencies with other system components. We argue that these two aspects represent the building blocks towards the ultimate security of system's access, and therefore, studying them together leads to the highest achievable security of access.

This work is indeed motivated by a practical application of the presented evaluation method to analyze and quantify failure interdependency between assets and security controls in access control implementations. The initial version of this work briefly introduced the problem and was presented in [5].

The rest of this paper is organized as follows: Section 2 briefly demonstrates the related work. The proposed approach is described in Section 3. Section 4 shows the proposed modeling formalization, including used definitions and notation, Bayesian Network (BN) representation, and inference-based analysis. A case study is presented in Section 5, followed by a conclusion and further research directions in section 6.

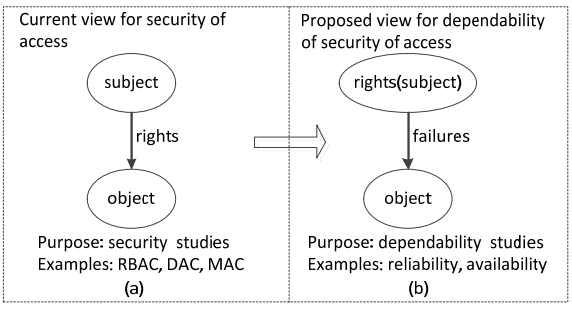

Figure 2. Current versus proposed view towards access controls. (a) the abstraction serving the purpose of security functions is centered on subject-object relationship, i.e., $\text{subject} \overset{\text{rights}}{\longleftrightarrow} \text{object}$**. (b) the abstraction serving the purpose of dependability functions is**

centered on asset-control relationship, i.e., object $\overset{\text{failures}}{\longleftrightarrow}$ rights(subject)**.**

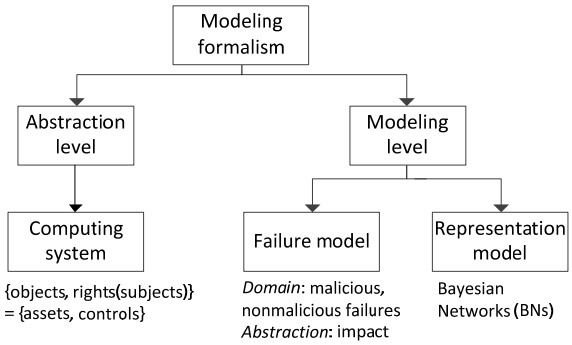

Figure 3. Key components in the proposed modeling approach.

2. Related work

Today, access control models receive a significant attention from both research and industrial communities [1], [2], [3], [6]. However, to the best of our knowledge, there is still a shortage of works addressing the dependability attributes of access control implementations (e.g., reliability, availability), not to mention the specific study of the impact of failure and its cascaded interdependency between assets and security controls.

The functional dependency, however, between subjects and objects is rather a different aspect; and it is studied from the access functions point of view. Thus, it is often handled by various techniques in the form of a separation of duties [2], [3]. As a result, the remedies rely on the type of the access control model in use. For instance, the RBAC framework presented in [3] administers security of access to system resources by the separation of the administration of RBAC from its access control functions, reducing dependency on security roles. Also, the work of [2] proposed a temporal RBAC model that introduces a new ways to control dependencies, by using temporal constraints to enable and disable different roles dynamically.

In addition to studying security functions offered by specific access control models, it is becoming clear that there is a need to analyze the operational ability of access controls themselves regardless of the class of access control model in use. Among existing probabilistic graphical methods, BNs in particular have shown to provide useful modeling and analysis results for capturing failure dependency relationships in complex systems [7], [8], [9]. Such a direction of research on BNs has even led to its connection to dependability studies [10], [11], the central field of failure studies.

3. Approach

For access control models, a computing system is abstracted into a set of subjects and a set of objects, with a set of rights determining the permissions allowed for subjects over objects [1], as demonstrated in Fig. 2a. To facilitate the dependability study of access controls, we propose the abstraction of the computing system into a set of assets and a set of controls. In this sense, objects (knowing that subjects can themselves be objects) are mapped into assets; and rights of subjects (in the form of implemented mechanisms) are mapped into controls (Fig. 2b). We then set the right failure model whereby plausible statistics can be established. Following the failure model, we use BNs modeling method to discover and evaluate the dependencies found among asset and access control components. This approach provides us with principled inference, reasoning and answers to various mission-critical queries about the security system. The proposed work, however, intersects with three fields of study: access control models, dependability theory, and graph theory, as demonstrated in Fig. 3.

4. Model formalization

4.1. Definitions and notation

To adapt BNs representation to the proposed level of abstraction and analysis, we set the following notation and definitions.

Table 1. Used notation.

Notation	Definition
S	$\{S_1, S_2, \dots, S_n\}$ system $\boldsymbol{S}$ of n components
A	$\{A_i: A_i \; is \; a \; an \; asset\}$ set of assets
C	$\{C_i: C_i \; is \; a \; control\}$ set of access controls
V	$\{Assets$ **A**, $Access \; controls$ **C**$\}$
E	$\{failure \; or \; breach \; dependency\}$
G	$(\boldsymbol{V}, \boldsymbol{E})$ Graph **G** of **V** nodes and **E** edges
P	probability distribution over **V** for failure dependency
X	$\{X_1, X_2 \dots, X_n\}$ system random variables
X_i	r.v. representing the state of failure of node i
$P(x_i)$	$P\{X_i = x_i\}$, probability of failure of node i
$pa(X_i)$	set of parents of X_i in **G**
$dec(X_i)$	set of descendants of X_i in **G**
$tag(A_i)$	$(Val(A_i), Avl(A_i))$
$Val(A_i)$	value of asset A_i
$Avl(A_i)$	availability function of asset A_i
$tag(C_i)$	$(Cst(C_i), Gol(C_i), Ser(C_i), Avl(C_i))$

Notation	Definition
$Cst(C_i)$	cost of control C_i
$Gol(C_i)$	security goal of control C_i
$Ser(C_i)$	security service of control C_i
$Avl(C_i)$	availability function of control C_i

The term *subject* is defined as an active entity, generally in the form of a person, process, or device, that causes information to flow among objects or changes the system state. *Object* is a passive entity that contains or receives information. *Resource* can be anything used or consumed while performing a function. The categories of resources are time, information, objects, or processors [12]. *Assets* can be anything that has value to the organization, its business operations and their continuity [13]. *Access* is a specific type of interaction between a subject and an object that results in the flow of information from one to the other. *Access control* is the process of limiting access to the resources of a system only to authorized entities [14]. Access control can be implemented by various mechanisms in the form of hardware or software components, operating or management procedures, or combinations of these. These mechanisms provide permission interfaces that mediate access of subjects to objects. In the context of this work, we use *controls* or *access controls* to implicitly denote access control mechanisms, not the process. Fig. 4 demonstrates these main entities involved in common access control architectures.

The *failure* definition adapted in this work is a representation based originally on the concept of failure in conventional reliability. The definition reflects the *deviation from correct service* [15], with the assertion to include failures from normal operational use and malicious activities on security systems [16], [17]. In addition, the level of abstraction of failure is the impact or consequence, not the underlying failure details [18]. We restrict the definition to failures involving access controls, not other system failures. Thus, we propose the following specific definition of failure in access controls:

the deviation of activities of legitimate subjects, or, alternatively,

illegitimate activities of legitimate subjects.

This definition is central to the modeling approach and analysis we propose.

Asset tag: A label associated with each asset to represent certain attributes of interest to the analysis. In this paper, the tag of asset i is represented by

$$tag(A_i) = (Val(A_i), Avl(A_i)),$$

where the term $Val(A_i) \in \mathcal{R}^+$ is the asset value, represented in countable units, say monetary units, and $Avl(A_i) \in [0,1]$ is availability measure, defined by the fraction of time asset A_i is functioning, or

$$Avl(A_i) = \frac{A_i \ up \ time}{A_i \ operational \ cycle}.$$

Control tag: A label associated with each control to represent its attributes of interest. The tag of control i is represented by

$$tag(C_i) = (Cst(C_i), Gol(C_i), Ser(C_i), Avl(C_i)),$$

where $Cst(C_i) \in \mathcal{R}^+$ is control cost, represented in the same unit as that used for asset value. The control goal $Gol(C_i) \in$ *{Physical security PGol, Computer security CGol}*, control service $Ser(C_i) \in$ *{Identification and authentication ISer, Authorization ZSer, Accountability ASer}*, and similarly, availability measure $Avl(C_i) \in [0,1]$.

These tags can be compiled during any assessment or estimation exercise on the system of interest, for example, during early phases of security risk assessment.

4.2. BN representation

A BN is a probabilistic graphical model that represents conditional dependencies among a set of random variables using directed acyclic graph (DAG). It is used for performing various probabilistic inferences. The representation of BN consists of two components. The first component, $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$, is a directed acyclic graph whose vertices[1] $\boldsymbol{V}$ corresponds to the random variables $X_1, X_2, \ldots, X_n$, which can be discrete or continuous. Graph edges $\boldsymbol{E}$ represent the relationships among these random variables, defining conditional

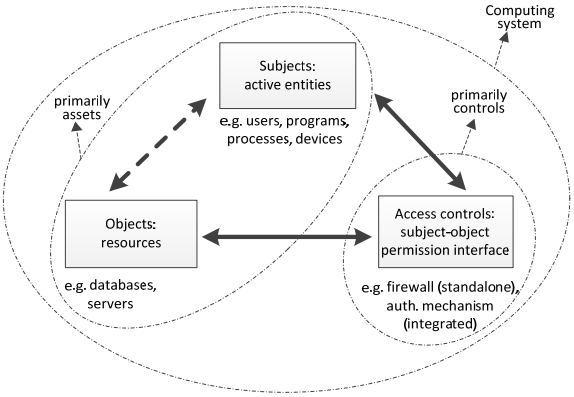

Figure 4. Typical entities of access control models in a computing system. Both subjects and objects interact directly with access controls (denoted by solid lines) to establish their indirect interaction with each other (denoted by dotted liens) [5]**.**

[1] Vertices and nodes are used interchangeably.

probability statements. The second component, $\boldsymbol{P}$, is the probability distribution over $\boldsymbol{V}$, defining a conditional distribution for each variable, given its parents in $\boldsymbol{G}$ [19].

Consider a BN represented by the finite set $\boldsymbol{X} = \{X_1, X_2, \dots, X_n\}$ of random variables with respect to $\boldsymbol{G}$. Each variable X_i may take value x_i from its domain[2]. The graph $\boldsymbol{G}$ encodes conditional independence assumptions, which allow the decomposition of any joint distribution into the product form using the chain rule [8], i.e.,

$$\mathrm{P}(\mathrm{X}_1, \mathrm{X}_2, \dots, \mathrm{X}_n) = \prod_{i=1}^{n} \mathrm{P}(\mathrm{X}_i / \mathrm{pa}(\mathrm{X}_i))$$

Also, $\boldsymbol{X}$ satisfies the local Markov property, meaning that each variable is conditionally independent of its nondescendants given its parent variables [11], i.e.,

$$\mathrm{X}_v \perp \mathrm{X}_{V \setminus \mathrm{dec}(v)} | \mathrm{X}_{\mathrm{pa}(v)} \quad \text{for all } v \in \mathbf{V}$$

The main idea of asset-control BN is to employ this representation to probabilistically capture the topology of system configuration from the perspective of access control model and associated failure dependency among its components. To show this, consider system $\boldsymbol{S}$ of n components, i.e., $\boldsymbol{S} = \{S_1, S_2, \dots, S_n\}$, where S_i is an asset or control according to the abstraction mentioned earlier. $\boldsymbol{S}$ is modeled by $\boldsymbol{X}$, $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$, and $\boldsymbol{P}$ as follows:

1. $\mathbf{G}$ is a directed acyclic graph.
2. Vertices $\mathbf{V} = \{\text{Assets } \mathbf{A}, \text{Controls } \mathbf{C}\}$, represented by the set of random variables $\mathbf{X}$ that makes up the nodes of the network.
3. Edges $\mathbf{E} = \{\text{failure dependency}\}$ are defined by the failure dependency among BN nodes where a directed link reflects the probable impact of failure of the initial vertex on the terminal vertex, i.e., of a control or an asset on another control or asset.
4. $\mathbf{P}$ is conditional probability distribution over $\mathbf{V}$, quantifying the effect of the parents' failure on each node.

Moreover, additional feature space pertaining to various attributes of access controls is attached to the graph nodes. As denoted earlier, these attributes are called asset tags (e.g., asset value $Val(A_i)$, availability $Avl(A_i)$) and control tags (e.g., cost $Cst(C_i)$, goal $Gol(C_i)$, service $Ser(C_i)$, availability $Avl(C_i)$).

[2] We use capital letters, such as X_1, X_2, for variable names and lowercase letters, such as x_1, x_2, to denote specific values taken by those variables.

This formalism allows us to map qualitatively and quantitatively the dependency and impact of security failures among assets and controls onto BN topology.

4.3. Inference-based analysis

Because a BN is a complete model for the variables and their probabilistic relationships, it can be used to answer various queries that can be very useful to the design and evaluation of a system. The proposed modeling allows us to study both security and insecurity attributes of access controls. Examples on security-related attributes are: asset value, availability of assets and controls, and protection goals and security services of controls. Examples on insecurity-related attributes are: risk, probability of failure, and cost of controls. Both types of attributes are reflected onto the same model foundation. Furthermore, there are many inference tasks that can be facilitated using the proposed modeling approach, as follows:

Joint distribution queries. These queries involve calculating the joint probability table between a set of variables. A task of this type is solved using Markov property, and takes the form

$$\mathrm{P}(\mathrm{x}_1 \mathrm{x}_2 \dots \mathrm{x}_n) = \prod_{i=1}^{n} \mathrm{P}(\mathrm{x}_i / \mathrm{pa}(\mathrm{X}_i))$$

Evidence-based queries. The goal is to determine the distribution of non-evidence variables given some evidence of failure (or non-failure). Inference can be done from children to parents or vice versa. This type takes the form

$$\mathrm{P}(\mathrm{X}_{v_1} / \mathrm{X}_{v_2}) \quad v_1 \neq v_2;\ v_1, v_2 \in V$$

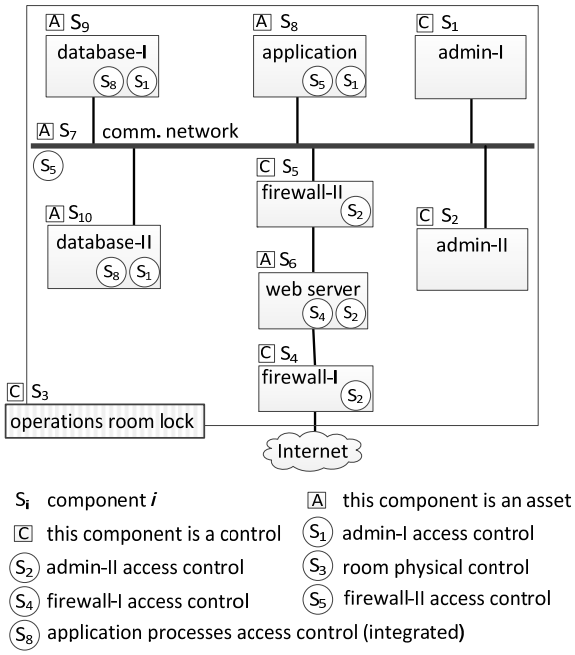

Figure 5. An outline of asset and access control entities in a simple web-based application model. Note that circles encode asset-control failure relationships.

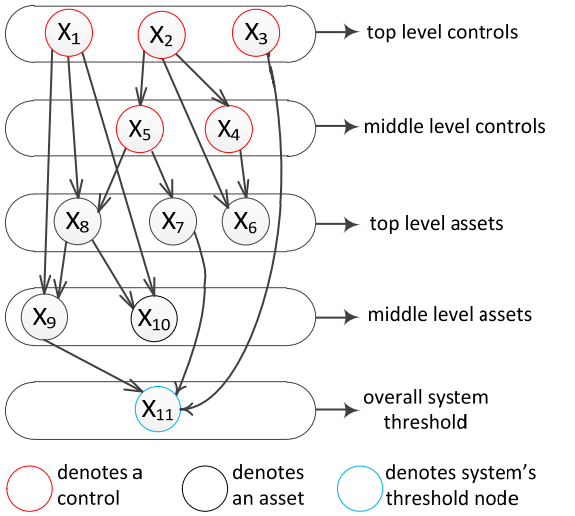

Figure 6. Bayesian Network representation of the asset-control graph example.

Independence check queries. The purpose here is to discover independency statements among different network nodes. This type usually involves conditioning on some variables to make such independence statements, taking the form

$$X_{v_1} \perp X_{v_2} / X_{v_3} \quad v_1 \neq v_2 \neq v_3;\ v_1, v_2, v_3 \in V$$

5. Numerical example

5.1. Case study

To simplify our analysis we will model typical access controls in common web-based application environments. One popular example is shown in Fig. 5 [5]. This scenario demonstrates a simple networked system of ten nodes representing a web-based application connecting two database platforms. For computer security, two admin platforms and two firewalls are used to control access to the system resources. For physical security, a lock mechanism is used to control access to the server farm. So, $\boldsymbol{S} = \{S_1, \ldots, S_{10}\}$.

Also, assume that expert knowledge concluded the following failure relationships for access controls: the failure of admin-I platform (S_1) leads to the failure of the application platform (S_8) and both database platforms (S_9, S_{10}). The failure of admin-II platform (S_2) impacts the failure of the web server (S_6) and both firewalls (S_4, S_5). The failure of the server farm's lock (S_3) breaches the whole system. Also, any failure on the application platform (S_8) causes disruptions to both databases (S_9, S_{10}), and the failure of database-I platform (S_9) leads to compromising overall system sensitive data. The failure of internal firewall, firewall-II (S_5), impacts the failure of the application platform (S_8) and communication network (S_7), and the failure of the communication network (S_7) impacts overall system operations. Finally, the failure of the external firewall, firewall-I (S_4), directly compromises the web server (S_6).

5.2. BN model representation

The corresponding BN topology is mapped directly using the assumed failure relationships in this scenario, as shown in Fig. 6. Assume the model random variables are binary, mutually exclusive and collectively exhaustive states of the probability of the failure space. $\boldsymbol{S}$ is modeled by $\boldsymbol{X}$, $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$, and $\boldsymbol{P}$ as follows:

1. $\mathbf{G}$ is a directed acyclic graph.
2. $\mathbf{X} = \{\mathrm{X}_1, \ldots, \mathrm{X}_{11}\}$,
   Vertices $\mathbf{V} = \{\text{Assets } \mathbf{A},\ \text{Controls } \mathbf{C}\}$, where
   $\mathbf{A} = \{\mathrm{X}_6, \mathrm{X}_7, \mathrm{X}_8, \mathrm{X}_9, \mathrm{X}_{10}, \mathrm{X}_{11}\}$, and
   $\mathbf{C} = \{\mathrm{X}_1, \mathrm{X}_2, \mathrm{X}_3, \mathrm{X}_4, \mathrm{X}_5\}$
3. Edges $\mathbf{E} = \{\text{failure dependency}\}$, where
   $\mathbf{E} = \{(\mathrm{X}_1, \mathrm{X}_8), (\mathrm{X}_1, \mathrm{X}_9), (\mathrm{X}_1, \mathrm{X}_{10}), (\mathrm{X}_2, \mathrm{X}_4), (\mathrm{X}_2, \mathrm{X}_5), (\mathrm{X}_2, \mathrm{X}_6), (\mathrm{X}_3, \mathrm{X}_{11}), (\mathrm{X}_4, \mathrm{X}_6), (\mathrm{X}_5, \mathrm{X}_7), (\mathrm{X}_5, \mathrm{X}_8), (\mathrm{X}_7, \mathrm{X}_{11}), (\mathrm{X}_8, \mathrm{X}_9), (\mathrm{X}_8, \mathrm{X}_{10}), (\mathrm{X}_9, \mathrm{X}_{11})\}$.
4. $\mathbf{P}$ is conditional probability distribution over $\mathbf{V}$, represented by the conditional probability tables

(CPTs) shown[3] in Table 2.

Also, assume the knowledge of the tags information, as compiled in Table 3, and finally assume that the embedded controls (controls built on top of other assets) are modeled combined with their assets to reduce the space of random variables.

It is remarkable that certain properties based on this BN representation arise. Particularly, the starting nodes are usually controls or unprotected assets. Middle nodes can be a mix of controls and assets. Leaf nodes are often assets as there is no point of having controls not protecting, thus not pointing to, any other assets. System is an added virtual node, depicted as a leaf node (X_{11} in Fig. 6). System node is used to represent the system-level failure threshold when main operations are considered in a failure state. This node is useful to define the scenarios and paths leading to system-level failures.

5.3. Analysis and Interpretation

Joint distribution queries. Different queries can be built to analyze different scenarios. In this part, we demonstrate how we can employ this approach to analyse the severity of failure of a particular node when combined with the failure of another node or a set of nodes. Then, we further use the BN representation to assess economic impact using the suggested asset and control tags. Recall that one of the main benefits of adopting BN approach in this work is bounding the failure topology, failure statistics, and economics of access control implementation onto the same model foundation.

For example, since the web server resides between two firewalls, we can set queries to analyze the operational and economic significance of failure of the web server (X_6) when combined with the failure of each of the two firewalls $\{(X_4), (X_5)\}$. First, to analyze the scenario that both the web server (X_6) and firewall-I (X_4) fail while none of the other controls is affected we compute the probability[4]

$$P(x_4 x_6 \neg x_1 \neg x_2 \neg x_3 \neg x_5)$$

$$= P(x_6/\neg x_2 x_4) P(\neg x_5/\neg x_2) P(x_4/\neg x_2)$$

$$P(\neg x_3) P(\neg x_2) P(\neg x_1)$$

$$= 0.30 \times 0.89 \times 0.09 \times 0.88 \times 0.90 \times 0.85$$

$$= 0.02.$$

This scenario raises the question about the possibility of either an accidental failure or external malicious activity on these two platforms.

Second, given the same asset-control failure relationships and statistics, to analyze the probability of failure of the web server (X_6) and firewall-II (X_5) while the other controls are functioning we write

$$P(x_5 x_6 \neg x_1 \neg x_2 \neg x_3 \neg x_4)$$

$$= P(x_6/\neg x_2 \neg x_4) P(x_5/\neg x_2) P(\neg x_4/\neg x_2)$$

$$P(\neg x_3) P(\neg x_2) P(\neg x_1)$$

Table 2. Hypothetical data for the conditional probability tables of the BN example.

$\mathbf{X_i}$	Conditional probability table (CPT)	
X_1	$p(x_1) = 0.15$	
X_2	$p(x_2) = 0.1$	
X_3	$p(x_3) = 0.12$	
X_4	$p(x_4/x_2) = 0.2,$	$p(x_4/\neg x_2) = 0.09$
X_5	$p(x_5/x_2) = 0.3,$	$p(x_5/\neg x_2) = 0.11$
X_6	$p(x_6/x_2 x_4) = 0.42,$ $p(x_6/\neg x_2 x_4) = 0.30,$	$p(x_6/x_2 \neg x_4) = 0.35$ $p(x_6/\neg x_2 \neg x_4) = 0.09$
X_7	$p(x_7/x_5) = 0.23,$	$p(x_7/\neg x_5) = 0.10$
X_8	$p(x_8/x_1 x_5) = 0.31,$ $p(x_8/\neg x_1 x_5) = 0.22,$	$p(x_8/x_1 \neg x_5) = 0.11$ $p(x_8/\neg x_1 \neg x_5) = 0.06$
X_9	$p(x_9/x_1 x_8) = 0.4,$ $p(x_9/\neg x_1 x_8) = 0.22,$	$p(x_9/x_1 \neg x_8) = 0.12$ $p(x_9/\neg x_1 \neg x_8) = 0.05$
X_{10}	$p(x_{10}/x_1 x_8) = 0.42,$ $p(x_{10}/\neg x_1 x_8) = 0.20,$	$p(x_{10}/x_1 \neg x_8) = 0.10$ $p(x_{10}/\neg x_1 \neg x_8) = 0.06$
X_{11}	$p(x_{11}/x_3 x_7 x_9) = 0.55,$ $p(x_{11}/x_3 \neg x_7 x_9) = 0.40,$ $p(x_{11}/\neg x_3 \neg x_7 x_9) = 0.18,$ $p(x_{11}/x_3 \neg x_7 \neg x_9) = 0.20,$	$p(x_{11}/\neg x_3 x_7 x_9) = 0.30,$ $p(x_{11}/x_3 x_7 \neg x_9) = 0.33,$ $p(x_{11}/\neg x_3 x_7 \neg x_9) = 0.10,$ $p(x_{11}/\neg x_3 \neg x_7 \neg x_9) = 0.05$

Table 3. Hypothetical data for the asset and control tags of the BN example.

$\mathbf{X_i}$	Description	Type	Tags: $\mathbf{tag(A_i)} = (\mathbf{Val(A_i)}, \mathbf{Avl(A_i)})$ $\mathbf{tag(C_i)} = \mathbf{Cst(C_i)}, \mathbf{Gol(C_i)}, \mathbf{Ser(C_i)}, \mathbf{Avl(C_i)})$
X_1	admin-I controls	control	(20, CGol, ISer, 0.95)
X_2	admin-II controls	control	(15, CGol, ISer, 0.95)
X_3	operations room physical lock	control	(6, PGol, ZSer, 0.99)
X_4	firewall-I	control	(18, CGol, ZSer, 0.98)
X_5	firewall-II	control	(12, CGol, ZSer, 0.97)
X_6	web server	asset	(19, 0.99)
X_7	comm. network	asset	(25, 0.999)
X_8	application	asset	(26, 0.95)
X_9	database-I	asset	(34, 0.98)
X_{10}	database-II	asset	(25, 0.98)
X_{11}	overall system	asset	(100, 0.96)

[3] Note that $P(\neg x_i) = 1 - P(x_i)$.

[4] We used BNT in Matlab for calculations. BNT is an open source toolbox, developed by Kevin Murphy, which implements several inference algorithms.

$= 0.09 \times 0.11 \times 0.91 \times 0.88 \times 0.90 \times 0.85$

$= 0.006.$

The result of these two scenarios suggests that when the web server fails the possibility of accidental failures or external malicious activity on firewall-I (X_4) is approximately three times larger than firewall-II (X_5). Their tags information suggests that these scenarios address the loss of access controls for computer security that provide authorization service. They also suggest that the cost ratio of firewall-I to the overall system controls is

$$\frac{Cst(X_4)}{Cst(\mathbf{C})} = \frac{Cst(X_4)}{\sum_{all\ C_i} Cst(C_i)}$$

$$= \frac{18}{71} = 25\%$$

and (X_4) domain of impact of failure is

$$\text{dec}(X_4) = \{X_6\}$$

with the value ratio of total impacted assets to the overall system assets

$$\frac{Val(\text{dec}(X_4))}{Val(\mathbf{A})} = \frac{Val(X_6)}{\sum_{all\ A_i} Val(A_i)}$$

$$= \frac{19}{229} = 8\%$$

In contrast, the cost ratio of firewall-II to the overall system controls is

$$\frac{Cst(X_5)}{Cst(\mathbf{C})} = \frac{12}{71}$$

$$= 17\%$$

and its domain of impact of failure is

$$\text{dec}(X_5) = \{X_7, X_8, X_9, X_{10}, X_{11}\}$$

with a value ratio of total impacted assets to the overall assets

$$\frac{Val(\text{dec}(X_5))}{Val(\mathbf{A})} = \frac{25 + 26 + 34 + 25 + 100}{229}$$

$$= \frac{210}{229} = 92\%$$

As shown, the analysis of the domain and impact of failure using BN topology and associated tags information augmented our findings. It shows that while firewall-I (X_4) costs more than firewall-II (X_5), it only contributes to mitigating the risk of failure on 8% of the value of total system assets. Conversely, firewall-II contributes to mitigating the risk of failure on 92% of total system assets. Table 4 shows a summarised statistics of these findings. Note that the second column represents the joint probability of failure of the web server (X_6) when combined with the failure of either of the two firewalls, while the other controls are functioning.

Evidence-based queries. Similarly, different set of queries can be built for various diagnosis and perdition exercises, analyzing both operational and economic significance.

For example on a diagnosis exercise, given a breach of the overall system (X_{11}), to find the probability that it was due to a failure initiated at admin-I controls (X_1), we basically find

$$P(x_1/x_{11})$$

$$= \frac{\sum_{x_2,x_3,x_5,x_7,x_8,x_9} P(x_1 x_{11} x_2 x_3 x_5 x_7 x_8 x_9)}{P(x_{11})} = 0.18$$

And by applying the same analysis for the rest of controls, we find

$$P(x_2/x_{11}) = 0.10$$

$$P(x_3/x_{11}) = 0.28$$

$$P(x_4/x_{11}) = 0.10$$

$$P(x_5/x_{11}) = 0.14$$

Table 4. A statistical summary of the joint-based query.

Node X_i	Joint prob. of failure	Control cost ratio $\frac{Cst(X_i)}{\sum_{all\ C_i} Cst(C_i)}$	Impacted nodes $dec(X_i)$	Impacted asset ratio $\frac{Val(dec(X_i))}{\sum_{all\ A_i} Val(A_i)}$
Firewall-I X_4	0.02	25%	$\{X_6\}$	8%
Firewall-II X_5	0.006	17%	$\{X_7, X_8, X_9, X_{10}, X_{11}\}$	92%

Table 5. A statistical summary of the evidence-based query.

Admin-I $P(x_1/x_{11})$	Admin-II $P(x_2/x_{11})$	Physical lock $P(x_3/x_{11})$	Firewall-I $P(x_4/x_{11})$	Firewall-II $P(x_5/x_{11})$
0.18	0.10	0.28	0.10	0.14

These figures show that the two most probable causes to system-level failures are: X_1 and X_3. Moreover, to predict system-level failures when these two controls fail, i.e., X_1, X_3, we write

$$P(x_{11}/x_1) = 0.11$$

$$P(x_{11}/x_3) = 0.21$$

In regards to causes and predictions of system-level (X_{11}) failures, these results, as summarised in Table 5, suggest that the operations room control (X_3) for physical security followed by admin-I (X_1) controls for computer security scored the highest among all controls.

Independence check queries. These queries allow us to study the independence exist in the relationships among different assets and controls. Obviously, the failures of the communication network (X_7) and application platform (X_8) are independent given we know the status of firewall-II (X_5), i.e.,

$$X_7 \perp X_8/X_5$$

Similarly,

$$X_9 \perp X_{10}/\{X_1, X_8\}$$

6. Conclusion and further research

The recent advances in computing and communication technologies have introduced new challenges that require new paradigms and models towards the study and design of reliable access control models.

The distinction and study of *subject-object* relationship with respect to *rights* has been a central step in the design and implementation of access control architectures. In this work, we propose the distinction and study of *object-rights(of subject)* relationship as a different, necessary abstraction for studying the dependability attributes of access controls. To this end, the idea of asset-control graph is proposed. It is an abstract, formal representation of the relationship between assets and controls in computing systems, capturing coverage and failure dependency in access control models.

Applying BN-based methods offers a wide range of analysis tools to the research problem. In addition to discovering interdependencies between system assets and controls, it allows us to analyze unforeseen threat scenarios resulting from the cascaded impact of failures. This method can also be used to engineer the requirements of access control models subject to economic constraints. Moreover, the adopted failure and associated BN-based modeling represent an extension of dependability studies into security studies and access control models in particular.

Further research might consider several avenues were not covered here: 1) extending the research to include building the BN and applying appropriate structure and parameter learning methods; 2) extending the inference analysis to cover a wider range of diagnosis and prediction inferences; and 3) modeling asset-control relationship allowing cycles, directed, and undirected graphs to model more asset-control behaviors.

Acknowledgment

This work is sponsored and funded by King Saud University, Riyadh, Saudi Arabia.

References

[1] R. S. Sandhu and P. Samarati, "Access control: principle and practice," *Communications Magazine, IEEE,* vol. 32, pp. 40-48, 1994.

[2] E. Bertino, P. A. Bonatti and E. Ferrari, "TRBAC: A temporal role-based access control model," *ACM Transactions on Information and System Security (TISSEC),* vol. 4, pp. 191-233, 2001.

[3] R. S. Sandhu, E. J. Coyne, H. L. Feinstein and C. E. Youman, "Role-based access control models," *Computer,* vol. 29, pp. 38-47, 1996.

[4] R. Anderson and T. Moore, "The Economics of Information Security," *Science,* vol. 314, pp. 610-613, 2006.

[5] S. S. Alaboodi and G. B. Agnew, "Bayesian networks for modeling failure dependency in access control models," in *Internet Security (WorldCIS), 2012 World Congress on*, Guelph, ON, Canada, 2012, pp. 176-182.

[6] N. Dimmock, A. Belokosztolszki, D. Eyers, J. Bacon and K. Moody, "Using trust and risk in role-based access control policies," in *Proceedings of the Ninth ACM Symposium on Access Control Models and Technologies*, 2004, pp. 156-162.

[7] A. Bobbio, L. Portinale, M. Minichino and E. Ciancamerla, "Improving the analysis of dependable systems by mapping fault trees into Bayesian networks," *Reliab. Eng. Syst. Saf.,* vol. 71, pp. 249-260, 2001.

[8] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach.* Prentice hall, 2010.

[9] M. Steinder and A. Sethi, "End-to-end service failure diagnosis using belief networks," in *Network Operations and Management Symposium, 2002. NOMS 2002. 2002 IEEE/IFIP*, 2002, pp. 375-390.

[10] R. E. Barlow, "Using influence diagrams," *Accelerated Life Testing and Experts' Opinions in Reliability*, pp. 145-150, 1988.

[11] R. G. Almond, "An extended example for testing Graphical Belief," *Statistical Science Research Report,* vol. 6, pp. 1-18, 1992.

[12] US Department of Defense, "DoD Trusted Computer System Evaluation Criteria," *DOD 5200. 28-STD, Washington, D. C. , US Department of Defense,* 1985.

[13] ISO/IEC 13335-1, "Information technology -- Security techniques -- Management of information and communications technology security -- Part 1: Concepts and models for information and communications technology security management," 2004.

[14] US Department of Defense, "Glossary of Computer Security Terms," *National Computer Security Center, NCSC-TG-004-88, Ft. Meade, Md.,* 1988.

[15] A. Avizienis, J. -. Laprie, B. Randell and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Transactions on Dependable and Secure Computing,* vol. 1, pp. 11-33, 2004.

[16] J. C. Laprie, "Dependability of computer systems: Concepts, limits, improvements," in *Software Reliability Engineering, 1995. Proceedings., Sixth International Symposium on,* 1995, pp. 2-11.

[17] B. Littlewood, S. Brocklehurst, N. Fenton, P. Mellor, S. Page, D. Wright, J. Dobson, J. McDermid and D. Gollmann, "Towards Operational Measures of Computer Security," *Journal of Computer Security,* vol. 2, pp. 3, 1993.

[18] D. M. Nicol, W. H. Sanders and K. S. Trivedi, "Model-based evaluation: from dependability to security," *IEEE Transactions on Dependable and Secure Computing,* vol. 1, pp. 48-65, 2004.

[19] N. Friedman, M. Linial, I. Nachman and D. Pe'er, "Using Bayesian networks to analyze expression data," *Journal of Computational Biology,* vol. 7, pp. 601-620, 2000.

