Risk Mitigation Decisions for IT Security

M. LISA YEO, Loyola University Maryland
ERIK ROLLAND, University of California, Merced
JACKIE REES ULMER, Purdue University
RAYMOND A. PATTERSON, University of Alberta

Enterprises must manage their information risk as part of their larger operational risk management program. Managers must choose how to control for such information risk. This paper defines the flow risk reduction problem and presents a formal model using a workflow framework. Three different control placement methods are introduced to solve the problem, and a comparative analysis is presented using a robust test set of 162 simulations. One year of simulated attacks is used to validate the quality of the solutions. We find that the math programming control placement method yields substantial improvements in terms of risk reduction and risk reduction on investment when compared to heuristics that would typically be used by managers to solve the problem. The contribution of this research is to provide managers with methods to substantially reduce information and security risks, while obtaining significantly better returns on their security investments. By using a workflow approach to control placement, which guides the manager to examine the entire infrastructure in a holistic manner, this research is unique in that it enables information risk to be examined strategically.

Categories and Subject Descriptors: K.6.5 [**Management of computing and information systems**]: Security and Protections—*Invasive software /and Unauthorized access*

General Terms: Security, Management

Additional Key Words and Phrases: Controls, Information Risk Management, Workflows

ACM Reference Format:
M. Lisa Yeo, Erik Rolland, Jackie Rees Ulmer, and Raymond A. Patterson, 2014. Risk Mitigation Decisions for IT Security. *ACM Trans. Manag. Inform. Syst.* V, N, Article A (January YYYY), 21 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION & LITERATURE

Enterprises are under increasing pressure to better manage operational risks, including information risks. As an example, in 2004, an accounts payable clerk used her computer to access her firm's accounting system and issued 127 checks payable to herself and others. Checks written were cashed or deposited into her account or the accounts of her accomplices. The clerk was able to alter the electronic check registers to make it appear as if the checks had been made payable to the firm's legitimate vendors. The firm lost at least $875,035. The clerk was caught, pleaded guilty to two counts of computer fraud and faced a maximum sentence of five years in prison and a $250,000 fine [DoJ 2004].

Author's addresses: L. Yeo, Information Systems and Operations Management, Sellinger School of Business and Management, Loyola University Maryland, Baltimore, MD 21210; E. Rolland, Ernest and Julio Gallo Management Program, University of California, Merced, CA 95343; J. Rees Ulmer, Krannert School of Management and CERIAS, Purdue University, West Lafayette, IN; R. Patterson, School of Business, Department of Accounting, Operations and Information Systems, University of Alberta, Edmonton, AB T6G 2R6.

DOI: http://dx.doi.org/10.1145/0000000.0000000

Headlines in influential media outlets routinely recall the latest information security breach affecting yet another organization. Unfortunately, the costs of such breaches add up to a significant amount of money. According to the Identity Theft Resource Center, there were 16,167,542 records reported as breached in 2010 [ITRC 2010]. If the estimates provided by the 2010 Annual Study: U.S. Cost of a Data Breach of $214 per record are close to accurate, the total cost in 2010 of data breaches is approximately $3.5 Billion in the United States alone [Ponemon Institute and Symantec 2011]. This estimate only accounts for breaches of confidential information, such as credit card numbers, social security numbers, drivers' license data, bank account numbers, etc. as this information is required to be reported to the state attorney general in most US jurisdictions. Firms are also growing increasingly aware of the value of informational assets and how attractive these assets could be to the wrong parties; assets such as patent applications, engineering designs, chemical formulations, corporate strategy documents, research and development documentation, among other potentially high-value information.

Numerous frameworks for managing risks to information and technology resources abound, ranging from the ISO series on risk management (ISO 31000, ISO 31010) and information security management (ISO 27000 series) to The Committee of Sponsoring Organizations of the Treadway Commission (COSO) to COBiT to the NIST standards for risk management and information security, and others. These standards and frameworks share many similarities in that information risks must be identified, assessed, and managed. Such risks are managed by making decisions on which risks to accept, which to transfer via sourcing agreements, insurance or both, and which to mitigate or reduce to a more acceptable level. Risks are typically mitigated by placing one or more controls at a specific step in a business process. A control might be a specific technology, for example an access control mechanism, or it might be a procedure, such as having a supervisor signature on an override. Controls also have varying degrees of reliability in terms of preventing or detecting erroneous or fraudulent data moving through a system. While each framework has strengths and weaknesses, each one defaults to a generic prescriptive approach, which can be more or less implemented as a type of systems checklist. Despite being generic and in theory, customizable to each organization's unique set of systems and processes, the checklist approach becomes extremely difficult for managers to use with today's complex arrays of processes and technologies.

The checklist approach falls short in at least two areas. First, workflows change over time, as do the threats. Appropriate controls may not be used for many reasons, such as the system complexity might be greater than anticipated by the creators of the checklist or the introduction of new technologies might limit the effectiveness controls. An example would be the introduction of a wireless access point in a warehouse management system by an employee outside of the IT organization. Second, managers might overspend or misallocate funds for controls because they are unable to assess the impact of the interaction between the controls available, potential attacks, and business processes. For example, an expensive control might be placed on a check printer which limits who can pick up a printed check. The printer might be located in a highly secured area which requires remodeling with expensive materials and a trained guard checks identification of those few employees allowed to print and pick up checks. However, if no background check is performed (a relatively inexpensive control) on the few employees allowed to print and pick up checks, then additional risk is introduced into the system despite the checklist.

The concept of the organization's workflows can be used to define the focus of security controls [Rodríguez et al. 2011]. Indeed, this was the motivation behind Section 404 of the 2002 Sarbanes-Oxley Act (SOX) in the US, which requires explicit manage-

ment of internal controls over financial reporting processes. By focusing on the ways in which people, data, documents, forms, processes, etc. interact to accomplish organizational goals, we can then make better decisions about which controls need to be placed in which workflow locations, in order to better manage the organization's overall information risk profile.

Not unlike physical sensor systems (for example, waterflow contamination detection systems [Watson et al. 2009]), multiple problems arise in selecting, placing, and managing internal controls for information risk management within organizational workflows. The problems of selecting and placing internal controls have long been addressed by heuristics, meaning that internal audit practitioners have developed checklists and guidelines for the selection and placement of such controls. The same is generally true for information security management. Good security managers follow prescriptive practices in selecting technology and policy controls typically generated by outside agencies and augmented by internal institutional experiences. While the checklist approach generally meets legislative requirements, this approach is likely sub-optimal from an enterprise information risk management perspective. Do the controls selected in the locations in which they are placed within the organizational workflow provide an optimal level of risk management? There is significant need to create an integrated, contextually holistic view of information risk management given the workflow processes of the organization.

The orientation of this paper is to develop decision models for managers to place controls, and then simulate the expected effectiveness of these controls against risk exposures. The goal of this research is to enable decision makers to integrate the analysis of controls into the workflow context. We formalize a representation of the investment and control placement problem within the overlapping and interconnected workflows of the organization, as well as propose insights and solutions to the problem. This work falls under the category of design science modeling; we model the organizational workflows and place controls to mitigate information risks. We test three solution methods to place controls, two of which are heuristics based on checklist-style decision methods. The third method uses an integer linear programming (IP) technique. We solve this problem with a budget constraint and then test the solutions with a period of simulated incident attacks. Depending on the controls selected, damages may or may not be mitigated. The incremental risk exposure of the three decision methods, compared to the lowest cost control expenditures, are used to evaluate relative effectiveness. This work is important because currently there is no method to effectively integrate information management risks within the context of the organizational workflow.

The rest of the paper is organized as follows. The literature review is in Section 2. We present our problem statement and formulation model in Section 3. Section 3 also describes two heuristic procedures for adopting controls. Section 4 presents the computational experiments and results, and Section 5 ends the paper with discussion and conclusions.

2. LITERATURE REVIEW

Earlier, we identified two central themes in risk management investment decisions; the need for both controls and an integrated view of risk in the context of workflows. In this section we review studies related to these themes starting with works related to making investments to manage information risk. This allows us to then present our model of control investment and placement within workflows in order to manage risk.

Gordon and Loeb [2002] proposed one of the earliest models for making economically rational information security investments. Their model takes into account the vulnerability of the information to be protected and the resources available to protect that information. They found that in certain scenarios, firms should only spend a fraction of

their expected losses to prevent security breaches, which is contrary to the popular belief at this time, which is that information security investments should be continually increasing. Bodin et al. [2005] incorporate the Analytic Hierarchy Process (AHP) into the earlier Gordon and Loeb [2002] model, in order to take advantage of qualitative information in making security investments.

Other researchers have since presented more complex and detailed models. Kumar et al. [2008] use a portfolio model of information security countermeasures to simulate the value of various portfolios against various attacks. They were able to demonstrate through simulation experiments that the interaction effects of the various security countermeasures can offer more protection to the organization than just the sum of each countermeasure's benefit, which indicates that an overall strong information security infrastructure can mitigate a weaker component of the infrastructure. Kumar et al. [2007] present an analytical model of investment decisions and countermeasures for protecting against availability and confidentiality-type attacks. Their model results in guidance to managers regarding investments and allocations to divisions for both availability-protecting and confidentiality protecting mechanisms. Herath and Herath [2008] propose a real options analysis (ROA) model for evaluating information security investments and present a Bayesian learning and post-audit function, in order to incorporate continuous information into their model. Cavusoglu et al. [2008] compare game-theoretic models to decision-theoretic models of security investment and report that game-theoretic approaches can result in better outcomes to the firm under certain conditions, which emphasizes the need to consider information security management a dynamic and strategic problem. While these papers take different approaches to modeling investments in information security, they all consider the interaction effects among security technologies, which is an important development in the literature.

The papers mentioned above generally tend to focus on security technologies, such as intrusion detection systems and anti-virus protection. The more general concepts of internal controls, which include access control technologies and internal audit processes, as well as technologies used to protect the confidentiality and integrity of data, have also been studied in the information security context. Researchers have examined the specific nature of controls used in protecting information systems. For example, Weber [1989] examined electronic funds transfer systems, and found a need to balance speed and ease of use with security. Wood [1990] prescribed twenty-three principles for designing controls in software, ranging from cost effectiveness to maintaining a low profile for the control.

Basu and Blanning [1997] define a workflow as "the flow of information and work through one or more organizational entities involved in business processes," [Basu and Blanning 1997, pp. 359-360]. Workflows are critical to organizations, as they depict the business processes and rules within the organization, and are necessary for systems analysis and design activities, as well as for efficiency and control purposes. Basu and Blanning [1997; Basu and Blanning [2000; Basu and Blanning [2003] propose using metagraphs as a formalization of organizational workflows, which allows for formal analysis of workflows and business processes.

Cernauskas and Tarantino [2009] suggested that combining business process management and process control can improve risk transparency and reduce operational losses. Kumar et al. [2008] examine different policies for countermeasure placement given information asymmetry between the CIO and division managers. In the context of auditing, Krishnan et al. [2005] provide a formal method of assessing data reliability that helps auditors choose the controls to review, balancing the cost and accuracy of assessment requirements. Their set covering model could be adapted to help answer the question about which controls to implement in order to reach a desired level of data reliability. Extensions to the work of Krishnan et al. [2005] provide a framework

for managing data quality risks in accounting information systems by modeling error propagation through the system, where the system is represented at the business process level [Bai et al. 2007; Bai et al. 2012b]. A Markov decision model then allows for the determination of the optimal control policy, for specific control procedures. This model assumes that at each error source, there is one control procedure that will be implemented, as opposed to selecting from a portfolio of controls, which could be a combination of technologies and procedures, each with varying costs and benefits.

To our knowledge there is scant literature which provides managers with support to establish the strategic placement of controls within workflows. Bai et al. [2012a] examine access control for information privacy and confidentiality within a workflow context. While the problem of access control is a critical and complex issue, our work examines the more general issue of control placement from an overall investment and risk management perspective.

Having identified the gap in the literature regarding the need for a model for strategic placement of controls, we formalize our problem statement in the next section. Following this, we present our model for optimal control placement within a workflow framework with the goal of mitigating information risk subject to budgetary constraints.

3. PROBLEM STATEMENT & MODEL DEVELOPMENT

The placement of controls is a matter of deciding on how to best guard against potential information security breaches given constraints. For a single workflow, there are multiple security scenarios that must be considered, each with multiple protection choices and implementation locations in order to address confidentiality, integrity, and availability concerns. Workflow controls have costs associated with their acquisition, implementation, and management. The multitude of choices in multiple scenarios becomes a combinatorial problem; multiple workflows amplify this combinatorial problem. Since most organizations will have clear constraints as to the budgets that can be spent on information security, we conclude that the resulting management problem is a combinatorial optimization problem with budget constraints.

Consider a standard purchasing workflow, as illustrated by Figure 1. We assume that a security breach may occur at any node in the workflow, although we recognize that some nodes might be more vulnerable than others. Our goal is to place high quality (efficient) controls that minimize the potential damage to data contained within the workflow.

In Figure 2 we present the workflow illustrated in Figure 1 simply as the set of nodes and edges to allow us to easily illustrate some hypothetical incidents such as breach of confidentiality (incident 1), loss of data integrity (incident 2), and impaired availability of data (incident 3). Incident 1 could be an intercepted electronic funds transfer (EFT). Incident 2 could be the deliberate altering of PO information. Incident 3 could be the loss of access to a database server due to power outage. To detect these incidents, many controls can be placed in many different locations, and the same control can even be placed in multiple locations. In general, the damage of an incident is lower the earlier we detect it. This decrease in damage, though, must be balanced by the cost of placing controls in multiple locations.

Some controls at particular locations will detect some but perhaps not all incidents. Control 1, placed at node 3, can detect incident 2 whereas control 2, placed at node 3, can detect incident 3. Control 1, placed at node 4, can detect both incidents 1 and 2. However, if we rely on control 1 placed at location 4 to detect incident 2, we will incur increased damages related to that incident compared to placing control 1 at location 3. Thus, it might be worthwhile placing control 1 at nodes 3 and 4, even though there is redundant coverage for incident 2. Thus, in Figure 2 we see that Control 1 has

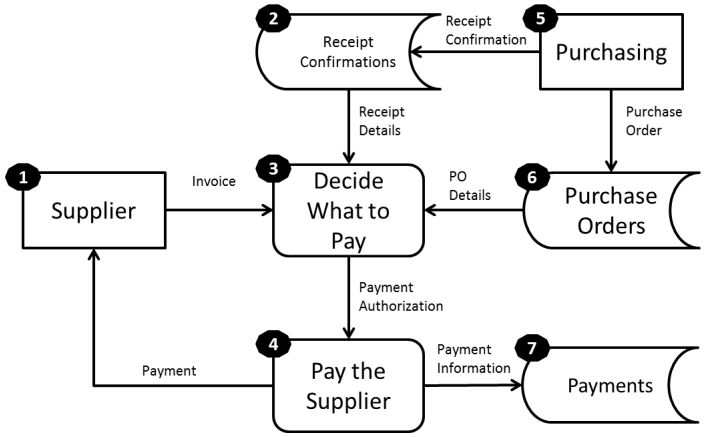

Fig. 1. Process Flow Example.

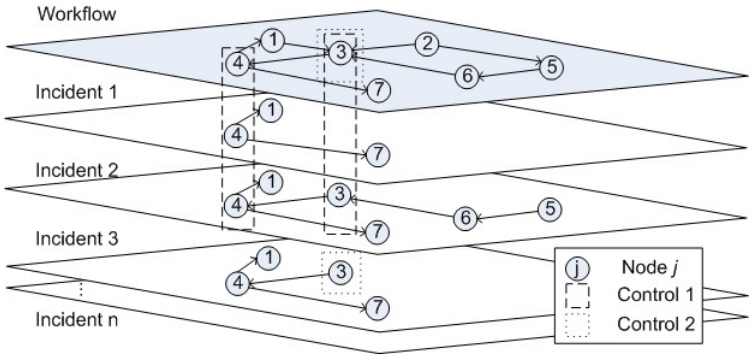

Fig. 2. Incident Examples.

been placed at nodes 3 and 4, and Control 2 has been placed at node 3 and all three incidents may be detected. Our proposed representation of this problem is restricted to the placing of at most one control for each incident on each path of the workflow. This means that managers must be prudent at incident scenario identification, and must also specify potential placement (or locations) for the controls.

3.1. Model

Risk reduction will occur through the strategic placement of controls within the workflow, given the costs and benefits of the controls under consideration, as well as the impact of the controls given the specific activities at each location within the workflow structure. We assume budget constraints, which limit the availability and effectiveness of the controls. Our approach is similar to the approach taken in the placement of sensors to detect contamination in water networks [Leskovec et al. 2007; Murray et al. 2009; Watson et al. 2009], but the types of breaches, workflows, and controls needed in our setting are more varied. Leskovec et al. [2007] also apply the approach to model the spread of information in blogs, identifying key blogs that quickly cover the majority of "information cascades." We can also apply this approach to contamination of information in organizational workflows.

Graph Theoretic Definition:

Given a graph with a set of location nodes (J) and edges (E), we define a set of incident scenarios (I) each describing an incident such as a security breach or the spread of unwanted data, and a set of controls (K) to mitigate incidents. An incident, $i \in I$, is initiated from a single node in the workflow, and spreads through the workflow in a pattern described by the incident tree formed by the incident i's connections to other affected nodes; that is, a set of arcs $\{\alpha, \beta\} \in E$. The collection of arcs $\{\alpha, \beta\}$ for an incident i depict a damage dissemination flow created by that incident. Incidents can be detected and controlled for by installing a control $k \in K$ at any affected location. For each location, $j \in J$, there are zero or more control options, where each control will apply to one or more of the incidents $i \in I$. The use of a control at a location would incur a cost that may be location dependent. For any location, we may elect to use zero or more controls to guard against each incident. Each control type used for an incident at a location implies a unique level of potential damage resulting from the incident. The flow associated with each incident is used as a proxy for the damage from an incident given its control location and type. Once an incident is detected at a location, we assume that all issues related to that incident are resolved. If, for some reason, this is not the case, then a separate incident must be constructed.

Data:

Incident and expected damage data are described by several variables. First, we must identify which incidents, $i \in I$ can effectively be controlled by a control $k \in K$ if it is placed at location $j \in J$. We store this information in the variable a_{ijk}, defined as follows:

$$a_{ijk} = \begin{cases} 1 & \text{if incident } i \text{ is covered by a control of type } k \text{ at location } j, \\ 0 & \text{otherwise} \end{cases}$$

Each control has a cost, c_{jk} associated with it and the cost may vary based on its location in the workflow. Each incident, if not detected, will cause the firm to incur an expected damage, D_i. However, by placing a control k at location j, this damage may be reduced by an amount d_{ijk}. As we assume that workflows may be described by spanning trees and that the first control along a path that can detect an incident will detect it, we must also keep track of the set of paths, P_i, for each incident. Note that a control can be placed at nodes where appropriate, and similarly restricted for some locations. That is, the a_{ijk} variable defines which controls (k) are useful for incident i at location j. So, if a particular variable a_{ijk} = 0 for an incident i at location j (for a specific control k) then that control k can't be placed at location j and be effective for incident i. Note also that this approach takes into account the non-Boolean and non-linear nature of this problem. For example, we may consider two incidents (A and B) which could occur alone or together, and then allowing this by creating separate relevant a_{ijk} entries as follows: A alone, B alone, and C (denoting A and B together). Thus, to appropriately accommodate such scenarios, every possible combination of incidents and control scenarios could be expressed in a_{ijk}.

A budget, B, limits the total monetary resources available to purchase (and presumably implement and manage) the controls. The goal of this paper is to select locations and types of controls in such a manner that the total expected damage is minimized, while complying with the budget constraint. Table I summarizes the notation used in our IP.

Decision Variables:

In our model for multiple coverage, there are two decision variables. It is possible to purchase multiple controls at each and every location in our workflows. Thus, we

Table I. Notation for IP formulation.

Term	Name	Description
I	Incidents	Set of incidents where $i \in I$
J	Nodes	Set of nodes within a workflow where $j \in J$
K	Controls	Set of controls where $k \in K$
B	Budget	Limit on amount to spend for controls
a_{ijk}	Applicability	Denotes which incidents i are controllable by control k at location j
c_{jk}	Control cost	Cost for deploying control k at location j
D_i	Uncontrolled damage	Damage of incident i with no controls
d_{ijk}	Damage reduction	Reduction in damage for incident i for deploying control k at location j
P_i	Paths	Set of paths defining incident i
s_{jk}	Selected controls	Decision variable
x_{ijk}	Incident controls	Decision variable

define:

$$s_{jk} = \begin{cases} 1 & \text{if control of type } k \text{ is implemented at location } j, \\ 0 & \text{otherwise} \end{cases}$$

We must also decide which of the controls purchased at each location will be used for detecting a given incident. Thus, we define:

$$x_{ijk} = \begin{cases} 1 & \text{if incident } i \text{ is covered by a control of type } k \text{ at location } j, \\ 0 & \text{otherwise} \end{cases}$$

Note that, in data generation, for every incident i we require $\sum_{j \in J} \sum_{k \in K} d_{ijk} \leq D_i$ so that damage is always non-negative, even when controls are used. We accomplish this by using the distance between nodes to calculate both D_i and the discounts, d_{ijk}. A full description of how the problem data are generated is provided in Appendix A.

Problem: Flow Risk Reduction (FRR)

Our solution is a two stage process where we first select controls to minimize damage. It is possible that different solutions, at different costs, can result in the same minimal damage. Thus, we perform a second stage where the value of the objective function from stage 1 becomes a constraint in stage 2 where we find the minimal cost solution.

In this formulation for stage 1, we want to maximize the reduction in damage associated with placing controls. This is equivalent to minimizing the total realized damage after placing controls as follows:

$$\min[\sum_{i \in I} D_i - \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} d_{ijk} x_{ijk}] \tag{1}$$

Subject to:
A breach is observed only if a control exists:

$$x_{ijk} \leq a_{ijk} s_{jk} \qquad \forall i \in I, j \in J, k \in K \tag{2}$$

For each incident i, at most one control is active on each path p in the set of paths P_i from the root node to each terminal node in the incident:

$$\sum_{j \in p} \sum_{k \in K} x_{ijk} \leq 1 \qquad \forall i \in I, p \in P_i \tag{3}$$

Total spending on controls must not exceed the budget amount.

$$\sum_{j \in J} \sum_{k \in K} c_{jk} s_{jk} \leq B \tag{4}$$

$$s_{jk} \in \{0,1\}, \quad x_{ijk} \in \{0,1\} \tag{5}$$

In this formulation for stage 2, we wish to minimize the total cost of controls with the constraint that the damages from this solution must not exceed the value of the objective function found in stage 1. Thus our new objective function becomes:

$$\min \sum_{j \in J} \sum_{k \in K} c_{jk} s_{jk} \tag{6}$$

Subject to:
Constraints (2) to (5) from stage 1
Total damage must not exceed the value of the objective function found in stage 1, Γ.

$$\sum_{i \in I} D_i - \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} d_{ijk} x_{ijk} \leq \Gamma \tag{7}$$

Note that for implementation purposes, a multiplier between 1.000004 to 1.000009 was used on Γ to accommodate for rounding errors in the solver.

The way this model is constructed recognizes the marginal benefits on controls, and will stop adding controls when the marginal cost exceeds the marginal benefit. An unlimited budget does not necessarily result in placing controls everywhere, as it would depend on the marginal returns for the controls. Thus, if large enough, the budget may not constrain the model.

The knapsack problem is NP-Complete [Karp 1972; 2010]. The FRR problem is a special case of the knapsack problem, where categories of items are available but you can pick at most one item from each category. These categories are the incidents in the FRR problem. The FRR problem is NP-Complete.

3.2. Heuristic Decision Making

Current management practice is to use checklists or heuristic rules of thumb to guide the placement of controls in workflows. We compare our formulation with two heuristic decision making models developed after several informal conversations with security managers and security consultants. Security managers do use a variety of approaches in formulating budget and spending plans. We were not able to find these approaches documented in the literature. In general, these budgets tend to be either a percentage of the previous year's budget, originally based on a set of "required" controls (essentially a checklist approach) plus any additional spending required to fend off unanticipated successful attacks, problems, and compliance issues or any combination of these three. A recent Forrester study reports that "CISOs use very few real financial models to support the budgeting process. ... CISOs use last year's budget to determine this year's budget." [Forrester Research Inc. 2013, p. 1]. Two heuristic decision models are developed to be representative of typical managerial decision processes for information security control placement and spending.

The first heuristic selects controls for locations that will result in the maximum reduction in damage across all incidents in an iterative manner. If the first choice of a control and location exceeds the budget, the heuristic will search through the remaining choices to see if there is a control at a location that can be afforded within the budget. It continues in this manner until the budget is reached or there are no more affordable controls.

(1) Calculate the discount across all incidents for deploying a control of type k at location j. That is, calculate $\sum_i a_{ijk} d_{ijk} \forall j \in J, k \in K$
(2) Select the control which results in the largest discount across all incidents. That is, set $s_{j^*k^*} = 1$ for the control k^* at location j^* which results in the largest value for $\sum_i a_{ij^*k^*} d_{ij^*k^*}$, if it fits within our budget.

Table II. Listing of variables for IP formulation.

Variable	Values
Incidents (I)	50, 100, 150
Locations (J)	50, 275, 500
Controls (K)	10, 15, 20
Budget Scale (BP)	0.05, 0.1
Maximum cost of control ($maxC$)	900, 950, 1000

   (a) If the first choice of control does not fit in the budget, look for the next best control that is affordable.

(3) Continue steps 1 and 2 until no more controls can be purchased within the budget.

The second heuristic determines the incident that has the largest expected damage and selects the control at a location that will minimize the damage for that incident. It then recalculates the expected damage for all incidents given the control and location selected, determines the incident with the highest remaining damages, selects the control at a location that minimizes the damage for that incident and repeats these steps until the entire budget is exhausted. Like the first heuristic, if the first choice of a control and location exceeds the budget, the heuristic will search through the remaining choices to see if there is a control at a location that can be afforded within the budget, stopping only once it cannot purchase any more controls within the budget.

(1) Select the incident, i', which has the highest expected damage if no controls are selected.
(2) Select the control k' at location j' which will result in the largest reduction in damage for incident i'. That is, find the largest value of $d_{i'jk}$ for this incident and set $s_{j'k'} = 1$, if it fits within our budget.
   (a) If the first choice of control does not fit in the budget, look for the next best control.
   (b) If no control for this incident fits within the budget, look at the incident with the next highest expected damage and repeat step 2.
(3) Recalculate the expected damages for all incidents given the control k' at location j' has been deployed. Thus, for any incident for which the control k' at location j' is effective, the expected damage should be adjusted by the discount, $d_{i'jk}$.
(4) Repeat steps 1 through 3 until no more controls can be purchased within the budget.

In summary, heuristic 1 first chooses the control that will result in the largest reduction in risk. Heuristic 2 tries to prevent the most severe threats (incidents). The IP solution method protects the most important systems against the most severe threats by placing controls that will result in the largest marginal benefit of a control.

Next, we test the effectiveness of each of these three control placement methods against a barrage of simulated attacks. The relative effectiveness of each method is measured as the reduction of risk achieved. The barrage of attacks, as outlined in Appendix B, will simulate a year of attacks with different distributions of realization.

4. COMPUTATIONAL EXPERIMENTS AND RESULTS

We created a collection of 162 unique data sets with randomly generated node locations, incidents and controls along with all other input data for the model defined in Section 3.1. This data set embodies the manager's expectations regarding attacks and control effectiveness as related to organizational workflows. The data generation is explained in detail in Appendix A. A summary of the data generation parameters is presented in Table II.

The IP control selection method and the two heuristics defined earlier are used to select a set of controls for each data set. We refer to the solutions obtained by each

Table III. CPU time to find controls, in seconds

	Average	**Minimum**	**Maximum**
IP stage 1	13.103	0.364	59.839
IP stage 2	1125.454	0.963	18794.844
IP total	1138.557	1.335	18846.140
H1 algorithm	0.012	0.001	0.064
H1 formulation	5.493	0.200	24.162
H1 total	5.506	0.201	24.232
H1a algorithm	0.012	0.001	0.042
H1a formulation	5.111	0.180	23.095
H1a total	5.123	0.181	23.137
H2 algorithm	0.126	0.007	0.665
H2 formulation	5.412	0.182	23.53
H2 total	5.538	0.190	24.092
H2a algorithm	0.028	0.001	0.153
H2a formulation	5.119	0.179	22.214
H2a total	5.147	0.180	22.329

method IP, H1 and H2, respectively. Algorithms H1 and H2 will spend the entire budgeted amount, BP. To create an apples-to-apples comparison between the IP method and the heuristics, we restrict the budget for the heuristic algorithms to match the cost obtained by the IP method, calling the two restricted budget methods H1a and H2a, respectively. In every case, the cost of the IP method was below the budget.

All experiments were conducted using MatLab and AMPL to call CPLEX 11.0.1 running on an IBM X-series 3550 with eight Intel Xenon processors running at 3.1GHz and 32GB of RAM. The operating system is Windows Server 2003 R2 Enterprise x64 Edition with Service Pack 2 applied.

Finding controls is a fairly efficient process. Each stage of this process is timed using MatLab's TIC and TOC stopwatch functions, which records elapsed wall clock time between the start and finish of the code segments described in Table III. The IP method takes the longest time to solve, taking on average 1,138.557 seconds (18.98 minutes). The worst case took 18,846.14 seconds (5.23 hours) to find the solution, with stage 2 consuming most of the time. Stage 1 finds a solution with the lowest expected damage in an average time of 13.103 seconds, and stage 2 finds the cheapest solution among all solutions with the lowest expected damage. In general, the number of node locations in the workflow has the most dramatic effect on the length of time it takes both stage 1 and stage 2 of the IP method to solve. In comparison, the heuristic methods take at most 24.232 seconds in the worst case and under 6 seconds on average. Note that the solution from each heuristic was input to a modified version of the formulation to efficiently calculate the damages; the s_{jk} values were input as data. The time to complete this step for each heuristic is recorded in the respective line labeled "formulation" in Table III.

Each data set effectively serves as a training set to create controls using the five alternative methods of choosing controls which will protect against future attacks. To determine if the control placements are effective, the system is placed under simulated attack. That is, to test the solutions given by the IP method and the heuristics under different attack scenarios, we generate a weekly set of attacks for an entire year (52 sets of attacks). Descriptions of the attack scenarios are provided in Appendix B.

The performance of each solution against attack simulation 1 which follows a uniform random distribution is presented in Table IV. To analyze the performance of each solution against the simulated attack scenarios, the reduction of risk for each solution is considered. The risk reduction measure is calculated as the difference between the

Table IV. RR and RROI for Attacks following Uniform Distribution

	RR					RROI				
	IP	H1	H1a	H2	H2a	IP	H1	H1a	H2	H2a
Average	100	97.9	81.8	88.7	62.3	100	3.9	81.6	3.5	62.2
Minimum	100	74.2	30.7	39.7	6.2	100	0.1	30.8	0.1	6.2
Maximum	100	100.9	99.4	100.0	97.7	100	51.6	99.4	51.2	97.7

Table V. RR and RROI for Attacks following Expected Distribution.

	RR					RROI				
	IP	H1	H1a	H2	H2a	IP	H1	H1a	H2	H2a
Average	100	97.7	80.1	88.3	60.4	100	3.9	79.9	3.4	60.3
Minimum	100	80.4	53.3	41.5	8.5	100	0.1	53.3	0.1	8.5
Maximum	100	100.6	96.4	100.2	89.8	100	50.7	96.4	50.1	89.8

damage of the suite of attacks when no controls have been placed versus the damage that is incurred under the solution of interest, where:

$$RR = \textit{damage without controls - damage given selected controls}$$

In Table IV, the average, minimum, and maximum risk reduction (RR) of each method as a percentage of the risk reduction found with the IP method is presented. The risk reduction on investment (RROI) is calculated by dividing the risk reduction by the cost of the solution solution method, where:

$$RROI = RR / \textit{cost of solution}$$

In Table IV, the average, minimum, and maximum RROI of each method as a percentage of the risk reduction on investment found with the IP method is presented. Thus, the values for the IP method are set to 100 in every instance since this is the baseline. A value below 100 indicates a worse solution, and a value over 100 indicates a better result than the IP method baseline.

The same results for Attack 2 which is also random, but follows the expected distribution, are presented in Table V. Results indicate that the IP method for finding controls is superior for both attack scenario 1 and 2 in terms of both risk reduction (RR) and risk reduction on investment (RROI).

Some general observations are made. Heuristic H1 may occasionally find the best solution under attack, and often finds solutions as good as IP, but at significantly higher cost. Heuristic H2 does not find the best solution, even at additional cost, but occasionally finds a better solution than the IP method. When constrained to the same cost as the IP stage 2 solution, the performance of both heuristic methods, H1a and H2a, deteriorate drastically in terms of both RR and RROI. Thus, we conclude that if the IP method is unsolvable due to excessive problem size, then the heuristics can be utilized to solve the problem. However, significantly poorer risk reductions are likely and the cost will be higher, driving the RROI down substantially compared to using the IP method.

The reason for the poor performance of the heuristic solutions H1 and H2 in terms of RROI is that they drastically overspend on controls while yielding slightly worse control configurations on average. The heuristics cannot make valid judgments with respect to when the marginal benefit of adding the next control outweighs its cost; the stopping rule for the heuristics is to continue until all available money is spent. As a result, while it is sometimes possible to increase the risk reduction (e.g., H1 occasionally finds a better solution than the IP in terms of RR), the cost of doing so makes the overall return on the investment much smaller than the IP's solution. Having said that, in some situations the overriding factor is risk reduction rather than RROI. An

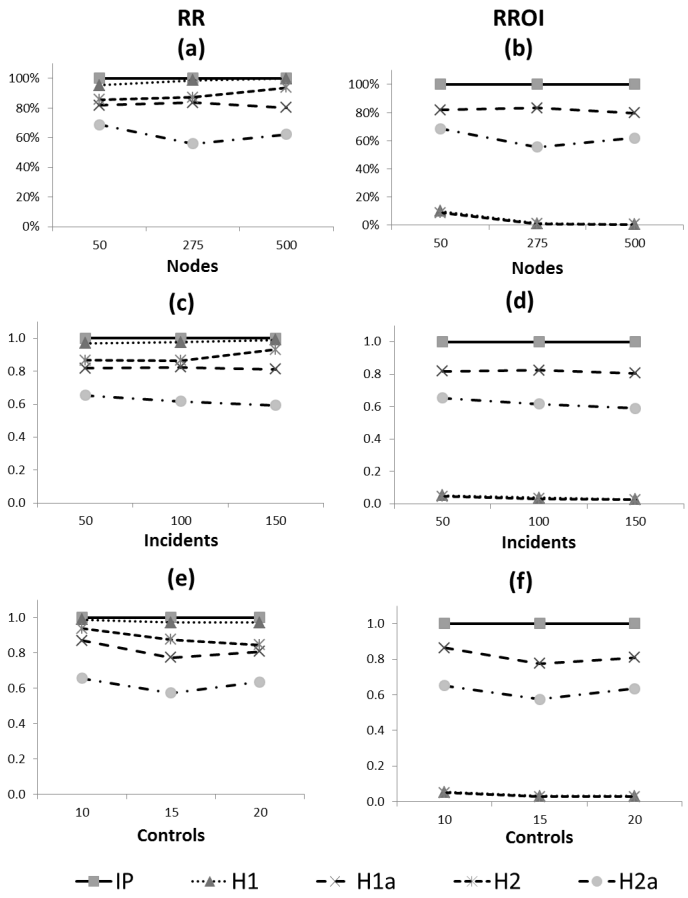

Fig. 3. Attack 1 results relative to the IP solution by Nodes (a, b), Incidents (c, d) and Controls (e, f).

example is when risk represents loss of life, and measuring the RROI of reducing the loss of life is considered to be unethical.

To compare how performance changes with the various methods as the parameter settings change, we illustrate the average performance under the simulated attacks for the number of node locations in Figure 3 (a) and (b), the number of incidents in Figure 3 (b) and (c), and the number of controls in Figure 3 (c) and (d). We see that the IP method is superior on average for all three parameters over all values tested. Figure 4 presents the equivalent results under the attacks simulated using the second method (Attack 2) presented in Appendix B.

Why doesn't the IP method produce the best set of controls to protect against attack in every situation, without exception? The IP method does produce the optimal solution for the expected incidents and attack frequency, which is why it performs so well across the board (note that all methods use the same expected values to produce control configurations). However, when the realized attacks are made, the actual attack occurrences in the experiment deviate randomly from the expected arrival rates. When reality differs from expectations, which is almost always the case, then the sub-

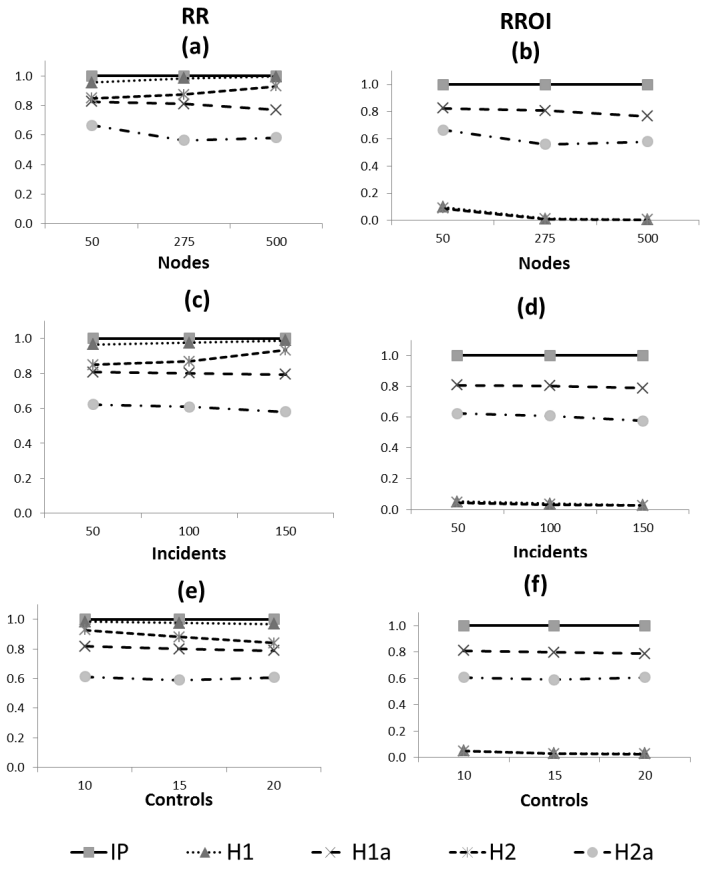

Fig. 4. Attack 2 results relative to the IP solution by Nodes (a, b), Incidents (c, d) and Controls (e, f).

optimal heuristics can sometimes produce better realized solutions than the "optimal" IP method. This is borne out in our experimental results.

We now consider the impact of budget reductions on algorithm performance. We utilize the cost of the IP solution (found in our initial experiments, and label this the initial IP solution value) as our initial budget constraint. We then solve problem FRR using methods IP, H1a, and H2a with budget constraints set at 100%, 90%, 80%, 70%, 60% and 50% of the initial IP solution value. We use only the H1a and H2a heuristics, as these are the 'budget constrained' versions of the heuristic algorithms. The IP solution method is superior in performance to H1a and H2a under all budget scenarios, but as expected, experiences a slight deterioration as the budget decreases to 50%. A comparison of the results at different budget levels is provided in Figure 5.

Next, we consider the case where we have imperfect information about possible future incidents. We define a hold back (HB) as an incident which is not included in planning, but which is a realized attack. We examine the impact of different levels of imperfect information, or hold back. The percentage of unknown HB events in the realized attack set is set to 0% (perfect information), 1%, 10%, 20%, 50% and 70%. We found that the IP method again produced the best results (as expected), but that the difference in the various solution methods decreased as the loss of information in-

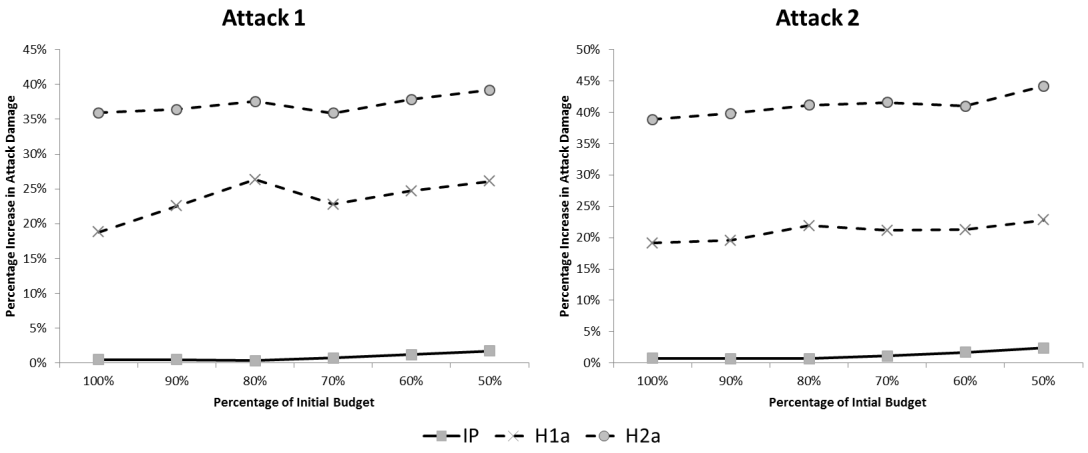

Fig. 5. Performance Deterioration from Best Known Solution with Budget Reductions.

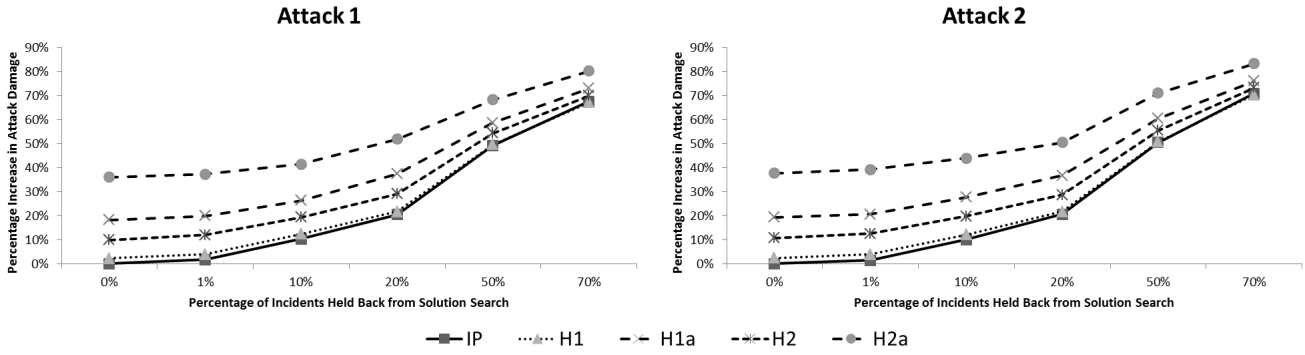

Fig. 6. Performance Deterioration from Best Known Solution with Unforeseen Events.

creases. A comparison of the results of different percentages of HB attacks is provided in Figures 6 and 7.

In summary, we can say that the IP method is the overwhelmingly best overall approach regardless of budget if the problem is small enough to be solved with this procedure. Having said that, we found no instances where the IP method failed to solve the problem, and we tested very large problem sizes of up to 500 control locations, 150 incidents, and 20 controls. What if solutions using the IP method cannot be found because the number of node locations becomes too large? Then a solution heuristic would have to be used, and heuristic H1 is superior to H2. Reducing the budget of the heuristics to compensate for overspending, as is done in H1a and H2a, results in dramatically poorer solutions. Thus, if the heuristic procedures are utilized, then artificially lowering the budget to prevent overspending will result in substantially more risk being carried by the firm. Thus, based on the experimental results, we can conclude that the use of the heuristics is never appropriate in a low budget environment. Additionally, even in a high budget environment, the IP method results in less risk on average being carried by the firm, and the heuristics perform much worse in the worst case scenarios.

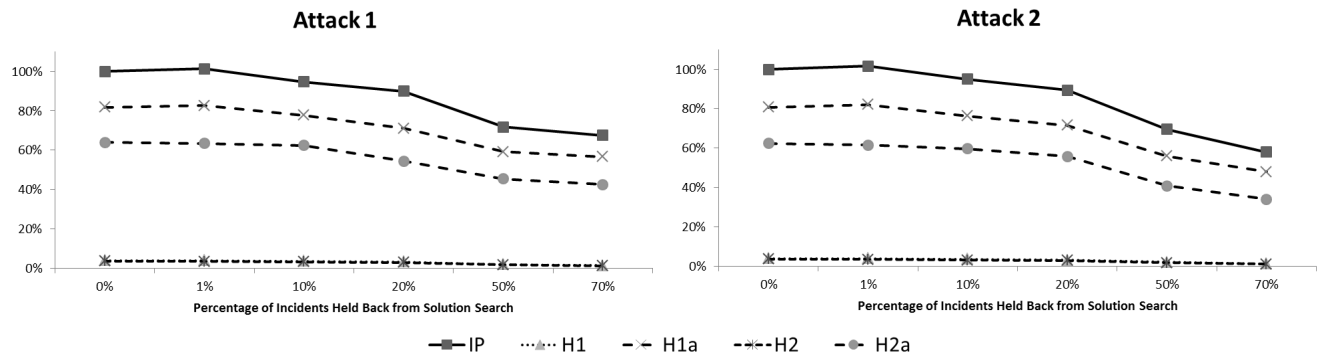

Fig. 7. Performance Deterioration in RROI with Unforeseen Events. We define the RROI for the IP solution with 0% HB as 100% and all other RROIs are reported relative this basis.

5. DISCUSSION AND CONCLUSIONS

Insights from this paper are twofold. First, we specify the FRR control placement decision problem using formal methods. This leads to a better understanding of the problem, and shows important connections between security investment decisions and information risk management outcomes. Second, we demonstrate how this problem can be solved using integer programming methods as well as heuristics. We demonstrate how trade-offs can be made with respect to security investments within the context of organizational workflows.

While the model currently assumes perfect detection of an incident by a control, it does not assume that the cost of all controls is the same. Future work could allow controls to detect incidents with probabilistic reliability and also could allow for differentiated damage prevention, meaning some controls are more effective than others against a given incident. However, our current model accommodates both the “worst case” and “expected value” views of the world.

The decision model allows for finding cost-efficient ways of protection against information security scenarios in the form of prevention or detection controls, or both - we do not make this distinction in the paper. The controls may have future impact on likelihoods of incidents, but this would have to be considered when preparing to solve the model again at some future time. Also, the model assumes that all control decisions are implemented in the current period. However, many control systems actually evolve over time and the decision to implement controls are made in the context of the control infrastructure that already exists and future controls that will take some time to implement, given the high time and cost to reallocate control resources from one task or location to another. For example, employees may need to be relocated to different cities, or new employees may need to be hired gradually over time. Future work can look at the multi-period dimensions of the problem, helping to identify not only optimal controls, but also optimal control implementation ordering.

There are multiple areas worth further exploration. More complex instantiations of controls, incidents, and workflows can all be considered, as mentioned previously. For example, a probabilistic solution approach could incorporate the use of overlapping controls at different locations, which would perhaps better detect breaches - particularly if those controls are imperfect (or probabilistic). Also, a stochastic modeling approach to this problem is potentially fruitful for future research. The concept of attack graphs could be incorporated into our workflow model to better characterize potential breach scenarios, as well as to identify potential controls and control points for

these scenarios in the workflow [Gupta and Winstead 2007; Peterson and Steven 2006; Phillips and Swiler 1998]. The model could also be extended to consider the additional risk reduction afforded by the purchase of so-called "cyberinsurance." The model could be further tested using actual incident data. Finally, given the extent of business process outsourcing, the model could be extended to examine cross-organizational workflows, building on work by Patterson et al. [2006].

The ultimate goal of this line of research is to build improved decision support tools for managers faced with managing the information risk in their enterprises. While the current practice of using intuition, experience, and best practices is an important starting point, managers who incorporate more formal methods, such as the proposed FRR model, can further improve resource allocation decisions and information risk management outcomes.

APPENDIX

A. DATA GENERATION

Here we describe the method used to generate 162 unique data sets for the parameter combinations listed in Table II. J node locations are randomly generated in a 100 by 100 space. We then randomly select an incident set, I, comprised of randomly selected subsets of nodes of a random size between 1 and J. The set of K controls is available to mitigate risk at each node in a the workflow. In addition, the cost of deploying a control, $k \in K$, at node $j \in J$ varies but is bounded by a value, $maxC$. Finally, the total budget for all controls is a percentage, $BudgetScale$ (BP), of total damages associated with the incidents when no controls are chosen. The data used for each iteration of the problem was generated in the following steps.

Nodes (Locations)

— Select coordinates in a 100×100 grid, randomly from a uniform distribution
— Calculate distances between each pair of nodes

Incidents

— For each incident, randomly select nodes to include. Each node has a 70% chance of being included in each incident.
— If an incident has no nodes chosen, randomly select one node for inclusion.
— For each incident, calculate the minimum spanning tree using Prim's algorithm and the distance matrix calculated in the node generation routine.
— For each incident, calculate every path in the spanning tree.
— For each incident, calculate the damage discount that would apply at each node if a control of type k was placed there, for all controls.
  — Calculate the total distance along all paths in the spanning tree.
  — For each node in the incident, calculate the total inflow to the node as the distance between the node and its upstream neighbour plus the total distance from the node to each endpoint in the node's path(s). See Figure 8 for an illustration.
  — The damage discount for each node is defined as the inflow to that node. That is, by placing a control k at node j, we are able to block damage from flowing any further and thus, discount the damage of incident i by the outflow distance at node j.

Incident-Node-Control Applicability array (a_{ijk})

For each incident, for each node, for each control, if the node is part of this incident, then randomly decide if this control will be applicable here (i.e. set $a_{ijk} = 1$) by randomly drawing from a uniform distribution. There is a 50% chance that the control will be applicable. Otherwise, set the $a_{ijk} = 0$.

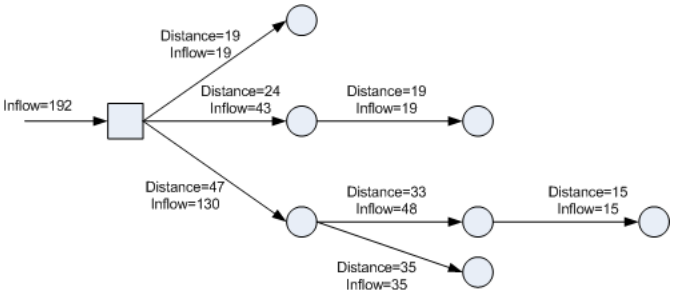

Fig. 8. Calculating inflow.

Table VI. Intervals used for generating uncontrolled damages.

Random Draw Interval	**Value Interval**	**P(incident i)**
$[0, 0.0001)$	[100 million, 10 billion]	0.0001
$[0.0001, 0.005)$	[10 million, 100 million]	0.0049
$[0.005, 0.05)$	[10,000, 10 million]	0.045
$[0.05, 0.95)$	[1000, 10,000]	0.90
$[0.95, 0.995)$	[10, 1000]	0.045
$[0.995, 0.9999)$	[0, 10]	0.0049
$[0.9999, 1)$	0	0.0001

Cost matrix (c_{jk}) For each control, for each location, randomly select an integer between 0 and the maximum cost of controls, $maxC$.

Discount matrix (d_{ijk}) The damage discount is the reduction in damage if the incident is detected at this node with this control. For each incident, node, and control combination, define the damage discount as the inflow from node j in incident i as calculated when generating the incidents as illustrated in Figure 8.

Total expected damages for incident (D_i)
In essence, we provide a structure to the damages such that rare incidents have extremely high (or extremely low) damages and the most common incidents have medium damages.

— For each incident, calculate the total, uncontrolled damage realized if this incident occurs:
  — Draw a random probability from a $\mathcal{U}(0, 1)$ distribution and record.
  — According to the break down in Table VI, add a value drawn from the appropriate interval to the total damage discount recorded when generating the discount matrix and set the probability that incident i will occur, P(incident i), accordingly.
— Sum uncontrolled damages across all incidents to use in calculating the budget for this data set
— Calculated the expected damages (D_i) for each incident by multiplying the uncontrolled damages of incident i by the probability that incident i will occur.

Budget Calculate the budget by dividing the total expected damages for all incidents divided by the number of controls in this data set then multiply by a budget scale (BP) defined for this data set. That is, $Budget = BP * \sum_{i \in I} E[D_i]/|K|$ where $|K|$ is the magnitude of K (i.e. number of controls).

B. ATTACK DESCRIPTION

B.1. Attack Simulation 1

In the first simulation, we select attacks from a uniform distribution with no consideration for the actual probability of seeing any particular incident.

(1) Randomly select an integer, n from $\mathcal{U}(0, I)$, of attacks for this period.
(2) Randomly select a set of n attacks from I, without replacement. That is, the same attack may not be seen more than once in any given period.
(3) Calculate the actual damage incurred for each solution, given this set of attacks has occurred.

We can then total the realized damages over the entire planning horizon of 52 periods and compare solutions.

B.2. Attack Simulation 2

The attacks in this simulation are drawn such that they follow the same probability distribution as the incidents.

(1) For each incident in I, if the probability of this incident is greater than or equal to a randomly drawn value, $\mathcal{U}(0, 1)$, add it to the set of attacks for this period. Once again, the same attack may not be seen more than once in any given period.
(2) Calculate the actual damage incurred for each solution, given this set of attacks has occurred.

We can then total the realized damages over the entire planning horizon of 52 periods and compare solutions.

C. BUDGET SHRINK

We used 27 unique data sets with randomly generated node locations, incidents and controls along with all other input data for the model defined in Section 3.1. These 27 data sets are a subset of the 162 data sets generated as explained in detail in Appendix A. A summary of the data parameters used in this experiment is presented in Table VII.

For each of the 27 data sets examined, we performed the following steps:

(1) Generate Attack 1 data.
(2) Generate Attack 2 data.
(3) Solve FRR using the IP as a priming run.
(4) Run Attacks 1 and 2 against each solution to determine damage in each case for the priming run.
(5) For Budget Reduction parameters of 0%, 10%, 20%, 30% 40%, and 50%
    (a) Multiply the *cost* of the priming IP solution by the budget reduction parameter
    (b) Solve IP, H1a, and H2a with the new budget restriction.
    (c) Run Attacks 1 and 2 against each solution

When analysing the results, we ignored the results from the priming run.

D. HOLD BACK

We used the first 100 unique data sets with randomly generated node locations, incidents and controls along with all other input data for the model defined in Section 3.1. These 100 data sets are a subset of the 162 data sets generated as explained in detail in Appendix A. A summary of the data parameters used in this experiment is presented in Table VIII.

For each of the 100 data sets examined, we performed the following steps:

Table VII. Listing of variables for data generation in the Budget Shrink experiment.

Variable	Values
Incidents (I)	50, 100, 150
Locations (J)	50, 275, 500
Controls (K)	10, 15, 20
Budget Scale (BP)	0.1
Maximum cost of control ($maxC$)	900

Table VIII. Listing of variables for data generation in the Hold Back experiment.

Variable	Values
Incidents (I)	50, 100, 150
Locations (J)	50, 275, 500
Controls (K)	10, 15, 20
Budget Scale (BP)	0.05, 0.1
Maximum cost of control ($maxC$)	900, 950

(1) Generate Attack 1 data
(2) Generate Attack 2 data
(3) For Hold Back parameters of 0%, 1%, 10%, 20%, 50%, and 70%
    (a) reduce the set of Incidents by the percentage of the Hold Back parameter
    (b) Solve IP, H1, H1a, H2, and H2a with the restricted set of Incidents.
    (c) Run Attacks 1 and 2 against each solution

This code resulted in 6 runs for each of the 100 data sets.

ACKNOWLEDGMENTS

This research was supported in part by a research grant from the CGA Association of Alberta. The data is available from the authors upon request.

REFERENCES

Xue Bai, Ram Gopal, Manuel Nunez, and Dmitry Zhdanov. 2012a. On the Prevention of Fraud and Privacy Exposure in Process Information Flow. *INFORMS Journal on Computing* 24, 3 (2012), 416–432.

Xue Bai, Manuel Nunez, and Jayant R. Kalagnanam. 2012b. Managing Data Quality Risk in Accounting Information Systems. *Information Systems Research* 23, 2 (2012), 453–473.

Xue Bai, Rema Padman, and Ramayya Krishnan. 2007. A Risk Management Approach to Business Process Design. In *Proceedings from the International Conference on Information Systems*. International Conference on Information Sytems.

Amit Basu and Robert Blanning. 1997. Metagraph transformations and workflow analysis. In *Proceedings of the Thirtieth Hawaii International Conference on System Sciences*, Vol. 4. 359–366.

Amit Basu and Robert W. Blanning. 2000. A Formal Approach to Workflow Analysis. *Information Systems Research* 11, 1 (2000), 17–36.

Amit Basu and Robert W. Blanning. 2003. Synthesis and Decomposition of Processes in Organizations. *Information Systems Research* 14, 4 (2003), 337–355.

Lawrence D. Bodin, Lawrence A. Gordon, and Martin A. Loeb. 2005. Evaluating Information Security Investments Using the Analytic Hierarchy Process. *Commun. ACM* 48, 2 (2005), 79–83.

Huseyin Cavusoglu, Srinivasan Raghunathan, and Wei T. Yue. 2008. Decision-Theoretic and Game-Theoretic Approaches to IT Security Investment. *Journal of Management Information Systems* 25, 2 (2008), 281–304.

Deborah Cernauskas and Anthony Tarantino. 2009. Operational risk management with process control and business process modeling. *Journal of Operational Risk* 4, 2 (2009), 3–17.

DoJ. 2004. Vallejo Woman Admits To Embezzling More Than $875,035. (2004). http://www.justice.gov/criminal/cybercrime/press-releases/2004/sabathiaPlea.htm

Forrester Research Inc. 2013. IT Leaders Should Look At Economic Value To Drive Security Spending Decisions That Protect Valuable Information Assets. (2013). https://www.verdasys.com/resources/forrester-it-economic-value-drives-security-spending-to-protect-information.pdf Last accessed May 10, 2013.

Lawrence A. Gordon and Martin P. Loeb. 2002. The economics of information security investment. *ACM Trans. Inf. Syst. Secur.* 5, 4 (2002), 438–457.

Suvajit Gupta and Joel Winstead. 2007. Using Attack Graphs to Design Systems. *Security Privacy, IEEE* 5, 4 (2007), 80–83. DOI:http://dx.doi.org/10.1109/MSP.2007.100

Hemantha S. B. Herath and Tejaswini C. Herath. 2008. Investments in Information Security: A Real Options Perspective with Bayesian Postaudit. *Journal of Management Information Systems* 25, 3 (2008), 337–375.

ITRC. 2010. 2010 Breach List. (2010). http://www.idtheftcenter.org/artman2/uploads/1/ITRC_Breach_Report_20101229.pdf

Richard M. Karp. 1972. Reducibility Among Combinatorial Problems. *Complexity of Computer Computations* (1972), 85–103.

Richard M. Karp. 2010. Reducibility Among Combinatorial Problems. In *50 Years of Integer Programming 1958-2008*, Michael Jünger, Thomas M. Liebling, Denis Naddef, George L. Nemhauser, William R. Pulleyblank, Gerhard Reinelt, Giovanni Rinaldi, and Laurence A. Wolsey (Eds.). Springer Berlin Heidelberg, 219–241.

Ramayya Krishnan, James Peters, Rema Padman, and David Kaplan. 2005. On Data Reliability Assessment in Accounting Information Systems. *INFORMATION SYSTEMS RESEARCH* 16, 3 (Sept. 2005), 307–326.

Ram Kumar, Sungjune Park, and Chandrasekar Subramaniam. 2008. Understanding the Value of Countermeasure Portfolios in Information Systems Security. *Journal of Management Information Systems* 25, 2 (2008), 241–279.

Vineet Kumar, Rahul Telang, and Tridas Mukhopadhyay. 2007. Optimally Securing Interconnected Information Systems and Assets. In *Workshop on the Economics of Information Security 2007*. Pittsburgh, PA.

Vineet Kumar, Rahul Telang, and Tridas Mukhopadhyay. 2008. Optimal Information Security Architecture for the Enterprise. *Working Paper-SSRN* (2008).

Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. 2007. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, San Jose, California, USA, 420–429.

Regan Murray, William E. Hart, Cynthia A. Phillips, Jonathan Berry, Erik G. Boman, Robert D. Carr, Lee Ann Riesen, Jean-Paul Watson, Terra Haxton, Jonathan G. Herrmann, Robert Janke, George Gray, Thomas Taxon, James G. Uber, and Kevin M. Morley. 2009. US Environmental Protection Agency Uses Operations Research to Reduce Contamination Risks in Drinking Water. *Interfaces* 39, 1 (2009), 57–68.

Raymond A. Patterson, Erik Rolland, and M. Lisa Yeo. 2006. Security and privacy in outsourcing with customer-specified risk tolerance. *eJETA* 2, 1 (2006).

Gunnar Peterson and John Steven. 2006. Defining Misuse within the Development Process. *Security Privacy, IEEE* 4, 6 (2006), 81–84. DOI:http://dx.doi.org/10.1109/MSP.2006.149

Cynthia Phillips and Laura Painton Swiler. 1998. A graph-based system for network-vulnerability analysis. In *Proceedings of the 1998 workshop on New security paradigms (NSPW '98)*. ACM, New York, NY, USA, 71–79. DOI:http://dx.doi.org/10.1145/310889.310919

Ponemon Institute and Symantec. 2011. 2010 Annual Study: U.S. Cost of a Data Breach. (2011). http://msisac.cisecurity.org/resources/reports/documents/symantec_ponemon_data_breach_costs_report2010.pdf

Alfonso Rodríguez, Eduardo Fernández-Medina, Juan Trujillo, and Mario Piattini. 2011. Secure business process model specification through a UML 2.0 activity diagram profile. *Decision Support Systems* 51, 3 (2011), 446–465.

Jean-Paul Watson, Regan Murray, and William E. Hart. 2009. Formulation and Optimization of Robust Sensor Placement Problems for Drinking Water Contamination Warning Systems. *Journal of Infrastructure Systems* 15, 4 (Dec. 2009), 330–339.

Ron Weber. 1989. Controls in Electronic Funds Transfer Systems: A Survey and Synthesis. *Computers and Security* 8 (1989), 123–137.

Charles Cresson Wood. 1990. Principles of Secure Information Systems Design. *Computers and Security* 9 (1990), 13–24.