LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-TR-635767

Decision Analysis Methods For the Analysis of Nuclear Terrorism Threats with Imperfect Information

J. C. Butler, P. M. Cronin, J. S. Dyer, T. A. Edmunds, R. M. Ward

April 26, 2013

Disclaimer

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

May 2013

McCombs Research Paper Series No. 2013-01

LLNL-TR-635767

Decision Analysis Methods For the Analysis of Nuclear Terrorism Threats with Imperfect Information

John C. Butler
McCombs School of Business
University of Texas at Austin
John.Butler2@mccombs.utexas.edu

Paul M. Cronin
McCombs School of Business
University of Texas at Austin
Paul.Cronin@phd.mccombs.utexas.edu

James S. Dyer
McCombs School of Business
University of Texas at Austin
Jim.Dyer@mccombs.utexas.edu

Thomas A. Edmunds
E Program
Lawrence Livermore National Laboratory
edmunds2@llnl.gov

Rebecca Ward
Kennedy School of Government
Harvard University
rebecca_ward@hks.harvard.edu

[THIS PAGE INTENTIONALLY LEFT BLANK]

TABLE OF CONTENTS

LIST OF FIGURES

[THIS PAGE INTENTIONALLY LEFT BLANK]

LIST OF TABLES

[THIS PAGE INTENTIONALLY LEFT BLANK]

LIST OF ACRONYMS

C/S	Containment and Surveillance
CCD	Chlorinated Cobalt Dicarbollide
CDF	Cumulative Density Function
CPLEX	optimization solver
DA	Destructive Analysis
DOE	U.S. Department of Energy
DOE-NE	U.S. Department of Energy - Office of Nuclear Energy
DPL	Decision Programming Language software
EPT	Extended Pearson-Tukey
GAMS	General Algebraic Model System software
IND	Improvised Nuclear Device
IP	Integer Programs
MFFF	Mixed Oxide Fuel Fabrication Facility
MIP	Mixed Integer Program
MIQP	Mixed Integer Quadratic Program
MOX	Mixed Oxide
NDA	Non-Destructive Assay techniques
NNSA	National Nuclear Security Administration
NORTA	Normal To Anything
NRC	Nuclear Regulatory Commission
PEG	Polyethylene Glycols
PIMS	Plutonium Inventory Measuring System
PRA	Probabilistic Risk Analysis
PRPP	Proliferation Resistance and Physical Protection Group
SMMS	Solution Measurement and Monitoring System
SNF	Spent Fuel
SNM	Special Nuclear Material
TALS-PEAK	Trivalent Actinide-Lanthanide Separation by Phosphorus reagent Extraction from Aqueous Complexes
TBP	Tributal Phosphate
TRU	Transuranic
TRUEX	Transuranic Extraction
UREX	Uranium Extraction

[THIS PAGE INTENTIONALLY LEFT BLANK]

EXECUTIVE SUMMARY

Protection of commercial nuclear fuel cycles against adversary groups who may attempt to steal weapons-usable material is an issue of interest to the U.S. Department of Energy, Office of Nuclear Energy. The wide array of potential nuclear fuel cycle configurations and the security measures that could be deployed to protect them pose a daunting system design problem. In general, there are billions of potential design configurations. Moreover, the design choices are not independent – the effectiveness of one design feature or security measure can affect the desirability of other design choices.

To explore this issue, we develop a mixed-integer optimization problem that solves a complex decision tree to allocate security resources across a commercial nuclear fuel cycle configuration. A nominal UREX+ reprocessing facility is used to demonstrate how the model allocates resources across a particular fuel cycle facility and shows how the problem can be scaled up to include other facilities and design features in the fuel cycle.

In our model, the defender first chooses a set of safeguards to implement in order to mitigate the risk of a terrorist theft from various unit operations in the reprocessing facility. The attacker observes the defender's strategy choice and executes an attack to acquire material, including the possibility of declining to attack at all. The model considers the terrorist as an intelligent adversary with a set of attack strategies known to the defender. The strategic choices made by both the defender and attacker affect the probability of a successful attack by the terrorist and in turn affect the distribution of consequences to the defender. We use a decision tree to illustrate the interactions among costs, decisions, correlated uncertainties, and outcomes, and then show how to formulate the problem as a mixed-integer program with an equivalent solution. The novel features of our formulation include the following.

First, as discussed above, we use a sequential game-theoretic formulation (a Stackelberg game). This formulation assumes the attacker can observe the defender's actions before the attack is executed. Many previous game-theoretic studies have assumed attacker and defender act simultaneously without knowledge of the other's strategy choice. We believe the sequential game formulation is more applicable for knowledgeable insider adversaries, the focus of this study.

Second, we build on our previous work that has developed a methodology to reflect the correlation among probability distributions represented as discrete chance nodes in a decision tree. We illustrate how this methodology can be applied to two chance nodes in a decision tree that relate the quality of the materials that might be stolen or diverted by an adversary with the yield of a weapon that might be constructed from that material. In a decision tree, the dependence of the yield on the quality of the material could be modeled with conditional probabilities of the weapon yield given the quality of the materials obtained. For three outcome discretizations, this would require the assessment of nine conditional probabilities from domain experts as part of the model building effort. In contrast, our methodology will allow this same representation of the tree based on the assessment of the correlation between these two chance nodes by the experts, which we believe is a much easier and more reliable assessment task. The advantage of this approach increases as the number of dependent chance nodes grows in a decision tree representation of this problem.

Third, we extend this approach to the approximation of the dependence between discrete binary events such as success or failure represented by binary chance nodes in a decision tree or in the special case of a probabilistic risk analysis (PRA) based on an event tree. This extension is a novel application in this domain, and uses Monte Carlo simulation to determine the appropriate conditional probabilities on the binary chance nodes when the number of correlated uncertainties is more than two.

Fourth, we show how these decision trees or probability trees can be formulated and solved as equivalent integer programming (IP) problems. The advantages of the IP representation include the opportunity to use the computing power associated with powerful optimization solver algorithms available on high speed computers to solve very large versions of these trees, and the ability to include several such models of different facilities into a much larger IP with connecting constraints that represent budget restrictions, number of available inspection opportunities in a given time window, or other dependencies among these facilities.

Policy makers can use the resulting model to inform their decisions about how to assess terrorism risk and to safeguard the commercial nuclear fuel cycle.

1 INTRODUCTION

The large number of potential domestic sources of nuclear materials in current and future domestic nuclear fuel cycle designs presents a challenge to fuel cycle and security system designers A methodology that could identify worst-case theft scenarios would allow counterterrorism efforts to focus on sources and material acquisition steps that a highly capable adversary is likely to use. Analytical methods to identify effective countermeasures against these scenarios would also help to manage this risk. These countermeasures could include reconfiguration of fuel cycle unit operations, intelligence collection assets, security at domestic nuclear installations, domestic checkpoints, radiation detectors, and other means to prevent theft of nuclear material, or to interdict an adversary.

The objective of this research is to develop integrated models of adversary behavior, countermeasure effectiveness, and system design that can be used to guide nuclear fuel cycle design efforts. The model must take into account adaptive (optimizing) adversary behavior and must provide a means for automated search through a large space of potential fuel cycle designs to find designs that are effective against such an adversary. Budget, logical, and other constraints on adversary behavior or fuel cycle design must also be satisfied. This work will focus on selected nuclear energy systems of current interest to DOE, and will demonstrate best practices and also highlight gaps in existing methodologies for modeling these problems, and thereby help focus R&D on improved methods.

Our work builds on previous efforts to combine aspects of game theory and traditional decision analysis, specifically decision trees, to take advantage of the concepts of the former and the practical implementation ease of the latter. We illustrate these methods utilizing a suite of nuclear terrorism threat scenarios directed at the domestic nuclear power infrastructure in which an adversary steals nuclear materials and subsequently fabricates an improvised nuclear device (IND).

1.1 TERRORISM THREATS

Nuclear fuel cycles must be protected against insider and outsider threats. Outsider threat groups would typically attack a facility using stealth and violence against barriers and personnel. The adversary must gain access to the material and exit the facility without being detected and interdicted by the guard force. Because many of the acts are overt, outsider protection is largely a matter of added security measures such as fences, motion sensors, cameras, and guards.

Insiders have authorized access to the facility and general knowledge of production processes, procedures, and security measures. In addition, some insiders have the authority to perform access control checks, material movements, or other critical functions. Protection against insiders requires effective design of unit operations, procedures, and material forms. Protection against the insider threat is generally considered more challenging. In this study, we focus on insiders stealing weapons-usable material in order to construct an IND, but the approach could easily be modified to analyze outside threats.

Detection and interdiction is based upon the system performance, which may not be a simple sum or product of performance measures of individual components. Furthermore, there is a need to assess performance of the system against all possible threat scenarios. Sometimes interactions or synergies among security components are present. For example, data from radiation detectors and surveillance video could be combined to track the movement of radioactive sources at a facility. This information could be checked with inventory control systems in order to detect unauthorized movement. To further assist in data interpretation, machine-learning techniques that learn normal behaviors of personnel and material and then flag anomalous behaviors might be used.

1.2 Previous Nuclear Fuel Cycle Evaluations

The Department of Energy's Office of Nuclear Energy (DOE-NE) has undertaken a methodical screening of hundreds of potential fuel cycle designs with a wide range of unit operations and material forms [DOE-NE 2012]. The project initially considered over 800 fuel cycle options, but aggregated them due to time constraints imposed on the study. The study developed a scoring system for the evaluation of fuel cycles with regard to a number of different objectives, including nuclear material security (i.e., protection against sub-national threats). They concluded that because physical protection can be added to any fuel cycle at a cost, the primary differentiator among fuel cycles is the material attractiveness at each stage of the process. Fuel cycles are evaluated on this basis.

The Proliferation Resistance and Physical Protection (PRPP) Evaluation Methodology Working Group has also been developing and applying methods for nuclear terrorism risk analysis [PRPP 2006]. The study describes the four physical protection vulnerability assessment steps as:

1. System element identification
2. Target identification and categorization
3. Pathway identification and refinement
4. Estimation of performance measures

Upgrades analysis is a fifth step that may be required if system effectiveness goals are not achieved. In this step, candidate upgrades to barriers, sensors, and procedures are evaluated to identify a system configuration that meets performance goals at minimum cost. This fifth step is the focus of the work described here. We focus on developing more detailed game-theoretic models of specific facilities and security systems, and this work is complementary to previous high-level evaluations in that regard.

Finally, additional studies of nuclear smuggling risks provide analytical tools that may be useful for assessing nuclear fuel cycle security. An overview of methods used for Homeland Security analysis is included in Maurer (2009). Network interdiction models are described in Wood (2011).

1.3 Example fuel cycle configuration, theft strategies, and security upgrades

In this study, we illustrate our algorithms using examples developed by Ward (2012) of "attacker and defender strategies" related to efforts by an insider (attacker) to acquire weapons-usable nuclear materials from a UREX+ aqueous reprocessing facility (defender). Ward (2012) has provided a diagram of the facility showing potential points of diversion. They are: diverting spent fuel from storage, diverting transuranic (TRU) material into the hulls, diverting material from any of the solvent extraction steps, or diverting TRU product from storage. Diversion of solution into hulls or from the extraction steps can be done with or without replacement with nitric acid to maintain mass and volume levels. For each of these points of diversion, she has identified potential actions that might be taken by an insider (attacker) to acquire nuclear materials and corresponding defensive actions that might be implemented by the facility managers (defender) to detect these threats.

Our primary focus is on a Stackelberg, or sequential-play, game formulation rather than a simultaneous-play Cournot game with Nash equilibrium. We believe the sequential play game with the defender forced to move first provides a useful perspective and analytical framework for an insider attacker who is afforded the opportunity to observe operations and security measures over long periods. In particular, it may be difficult to identify the potential insiders (attackers) and so the defender must move first in an attempt to provide some protection of the

facility and the adversary can choose to act after the defensive strategy of the system has been chosen. Solution algorithms for Stackelberg games have been developed by Bard (1998), Wood (2011), and Paruchuri, et al. (2008).

1.4 Example and Motivation

To motivate the problem and to illustrate the framework used in our methodology, we provide the following simple example with a limited number of decision nodes and uncertainties. The setup is a nominal nuclear fuel cycle facility. The defender (United States) makes two decisions on whether or not to implement two particular safeguards. The attacker, having – at least partially – observed the decisions made by the defender, chooses whether or not to attack the facility. If he attacks, it is uncertain whether the attack will be successful. If it is successful there is additional uncertainty about the quality of the material stolen during the attack and the nuclear yield that can be produced from the material. The decision tree as implemented by the commercial software package DPL is provided in Figure 1.

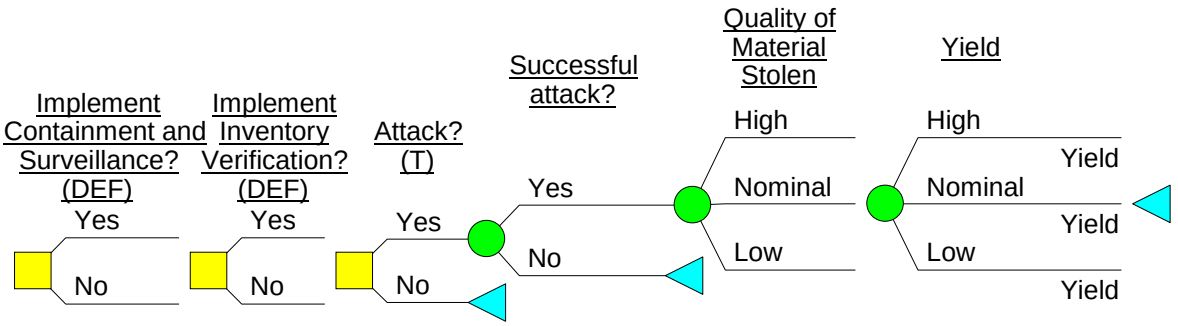


Figure 1: Example of a simple decision tree

It is important to note that the probabilities and consequence values used in the model are given merely for the purposes of illustration. They are not informed by any unclassified or classified information regarding the probability of an attack or the real-world effectiveness of any particular safeguard. The probabilities have been provided solely to make the examples easier to understand by the reader and to make the model operational. The reader should not infer anything further from these probabilities or values.

1.4.1 Decisions

The defender can choose to implement none, one, or both of the safeguards. In this example, the proposed alternative safeguards are to implement cameras and surveillance equipment and to implement an inventory verification system. These safeguards are chosen from the defender-attacker strategy matrix provided by Ward (2012) and will be further discussed later in the report. Implementing a safeguard increases the probability of detection of an attacker, and therefore decreases the attacker's probability of the successful theft or diversion of nuclear materials, but the defender incurs a cost. In our report we assume that detection is equivalent to interdiction; that is, if an attack is detected it is prevented.

After observing the defender's strategic choices, the attacker decides whether or not to attack. The outcomes of a successful attack are positive values representing a loss to the defender, so the defender wants to minimize the expected value at his decision node (smallest loss) whereas the attacker maximizes this expected value (largest value is best). The probability of a successful attack for the terrorist is determined by the defender's strategy. In this initial example the attacker and defender realize the same outcomes after the attack but the defender pays for each safeguard deployed and the attacker incurs a loss for an unsuccessful diversion.

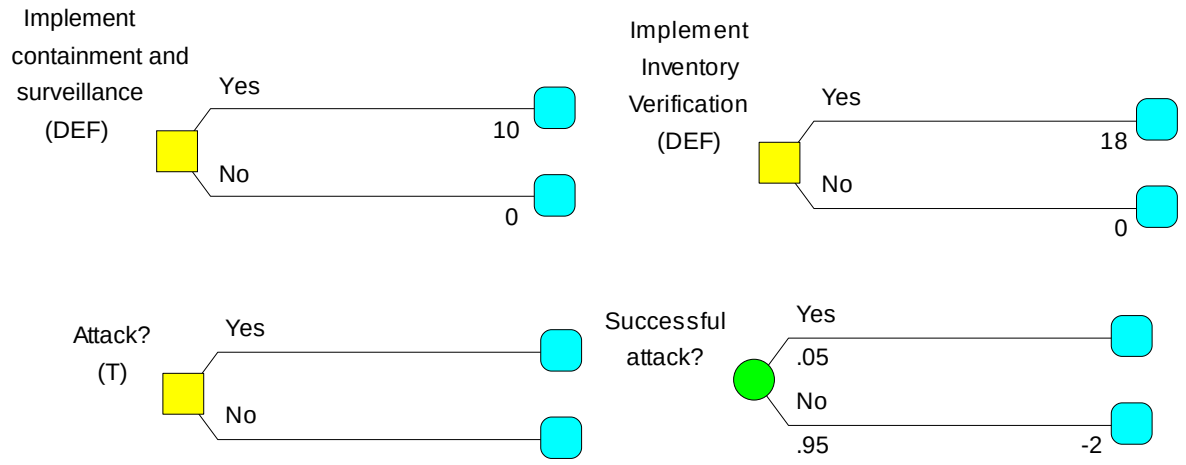


Figure 2: Defender and Attacker Strategies

The defender's decisions are pictured at the top of Figure 2. Choosing to use the cameras and surveillance safeguard costs the defender 10 whereas implementing the inventory verification safeguard is more expensive with a fixed cost of 18. These unitless costs are notional amounts used for illustration purposes only. The costs are additive; choosing both safeguards costs 28. Naturally, opting to not implement a safeguard does not incur a direct cost.

Costs are not considered in the decision node for the attacker. An argument could be made that there are some costs faced by the attacker in moving an attack plan forward. However, we feel those costs can best be captured as the impact of a failed attack and these costs are reflected in the outcome branch of the chance node corresponding to this possibility. For this example, we assumed that an unsuccessful attack was undesirable to the attacker since it could possibly: (1) reveal an insider if one existed, (2) possibly result in an intelligence gain to the U.S., (3) be a rallying event for the U.S. (i.e. we caught the terrorists), or (4) be a public setback for the terrorist group and its supporters. As a result, we assigned an arbitrary cost of 2 to an unsuccessful attack, which is shown in the chance node of Figure 2. In the case of a successful attack, the outcomes can account for any costs to initiate the attack.

Note that we have explicitly modeled the attacker's decision based on the assumption that the attacker has full knowledge of the defender's strategies and of the values of the outcomes. An alternative that could easily be incorporated into our models is to model the attacker's strategy with a chance node. This would reflect the defender's uncertainty about which strategies the defender has actually implemented, about the effectiveness of these defensive strategies, or uncertainty about the attacker's motivations and capabilities from the viewpoint of the defender. The pros and cons of modeling the attacker as a rational decision maker, versus taking a more probabilistic view of the attacker's actions, are discussed by Parnell, et al. (2010), and by Ezell and Collins (2010).

1.4.2 Uncertainties

The attacker has to consider not only the probability of being successful but also the probability distribution of the quality of the material stolen and the yield from any device that can potentially be built from that material. We first discuss the probability of the defender detecting an attack and the probability that the attacker would be successful. All device yield relationships described in this report are also notional.

1.4.2.1 PROBABILITIES OF DETECTION AND SUCCESS

An attack strategy is successful if and only if the attacker is undetected by all of the safeguard measures in place. The probability of successfully passing one safeguard undetected is $P_{success} = 1 - P_{detection}$. In this example, we consider the effectiveness of all safeguards to be independent. In other words, the effectiveness (or probability) of inventory verification in detecting an attack is independent of the effectiveness of cameras and surveillance in detecting an attack. If D is the set of all defender strategies, S is the set of all safeguards, and A is the set of all attack strategies then the probability of a successful attack by the terrorist is $P_{da} = \prod_{s \in S}(1 - P_{sa})$, where $a \in A$ is the chosen attack strategy, $s \in S$ is the defender's safeguard, and P_{sa} is the probability that action a is detected if safeguard s is implemented.

While the probability of detection for each safeguard is treated independently, it is important to point out that the probability of detection is conditional on all of the safeguards implemented. As shown in Figure 3, the notional probability of detecting a spent fuel rod theft is 0.8 if the cameras and surveillance strategy is implemented; otherwise, it is zero. Similarly, the assumed probability of detecting a spent fuel rod theft is 0.85 if the inventory verification strategy is implemented; otherwise, it is zero.

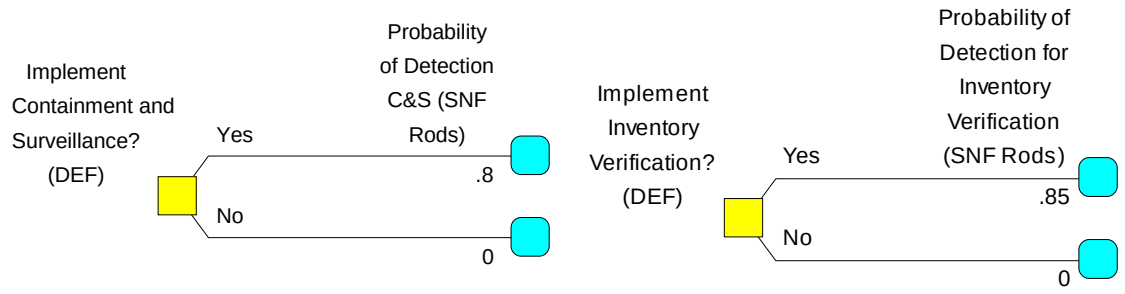


Figure 3: Conditional Probability of Detection

In this example, the attack set is $A = \{\text{spent fuel rod theft}, \text{no attack}\}$, and the defender set is $D = \{\text{cameras and surveillance}, \text{inventory verification}, \text{both}, \text{none}\}$. To show how the probability of success is calculated, consider the four possibilities in Table 1.

Table 1: Calculating the Probability of Successful Theft/Diversion

Implement C&S?	Implement Inventory Verification?	Probability of Detection for C&S	Probability of Detection for Inv. Ver.	Probability of Successful Theft/Diversion
Y	N	0.80	0.00	(1-0.8)(1-0) = 0.2
N	Y	0.00	0.85	(1-0)(1-0.85) = 0.15
Y	Y	0.80	0.85	(1-0.8)(1-0.85) = 0.03
N	N	0.00	0.00	(1-0)(1-0) = 1

Implementing both safeguards reduces the probability of success for the attacker but there is an additional cost incurred for the use of both safeguards. In the fourth case, it appears that if no safeguards are implemented the attacker will be successful with absolute certainty. In reality, there are a number of mandatory security guidelines and protocols that must be implemented in the operations of a nuclear fuel cycle facility. All such facilities likely have some safeguards in place including some baseline number of cameras and verification systems. The purpose of our model in this context is to show how additional safeguards – augmenting the baseline level of safeguards already mandated by the National Nuclear Security Administration (NNSA), Nuclear Regulatory Commission (NRC),

and other government agencies – can improve fuel cycle security in the face of terrorism risk. As a result, these probability calculations should be interpreted as marginal improvements relative to the status quo level of security.

1.4.2.2 Quality of Material Stolen and Yield

The next set of uncertainties relates to the consequences of a successful attack, which occurs in the tree only if the terrorist chooses to attack and is successful. For this simple example, we assume the notional value to the adversary of the yield of a nominal improvised nuclear device (IND) made from the stolen material is given by a continuous beta distribution with $\alpha = 2$ and $\beta = 4$ and bounds of $[50, 100]$. The attacker can take the form of many terrorist groups or actors, each of which may have different capabilities and knowledge. Terrorist acquisition of the material does not necessarily indicate the capability of the attacker to construct a weapon or some other device to be used against the defender. Yield can depend on multiple factors including the quality of the stolen material, but also the capabilities of the adversary to transform the material into another usable form for their purposes without being caught.

This example is hypothetical but the units of this yield distribution could be some measure of the explosive or destructive power of the device, or some assessment of the value a terrorist organization places on the successful development and deployment of the device. Then, we use the Extended Pearson-Tukey method (Keefer and Bodily 1983) to discretize the continuous beta distribution by assigning the probabilities of 0.185 to its 5th and 95th percentiles, and 0.63 to its median. For example, the 95th percentile of Beta (2,4) with bounds [50, 100] is 82.87. This approximation is shown in Figure 4 where the continuous beta distribution is pictured and the 5th, 50th, and 95th percentiles are labeled to correspond with the discrete representation on the right.

The same approach can be used to discretize the continuous distribution for the quality metric of the material stolen; assuming arbitrarily that this value is given by a $\mathrm{Beta}\,(4,3)$ distribution with bounds of [0,1]. This latter result is also shown in Figure 4. The distribution over the quality of the material stolen captures the uncertainty over nuclear material stability, quality, and other related factors that would be important for the transport, handling, and use of the material.

The material quality and yield distributions are shown separately; however, it seems plausible to assume that these distributions are not independent. A higher quality of nuclear or radiological material that is stolen should lead to a greater yield for an IND, on average. For example, spent fuel burn-ups are associated with the quality of material. A discussion of how to represent correlated uncertainties in decision trees is discussed in more depth in Section 4. However, to illustrate this simple example, we assume the distributions are independent.

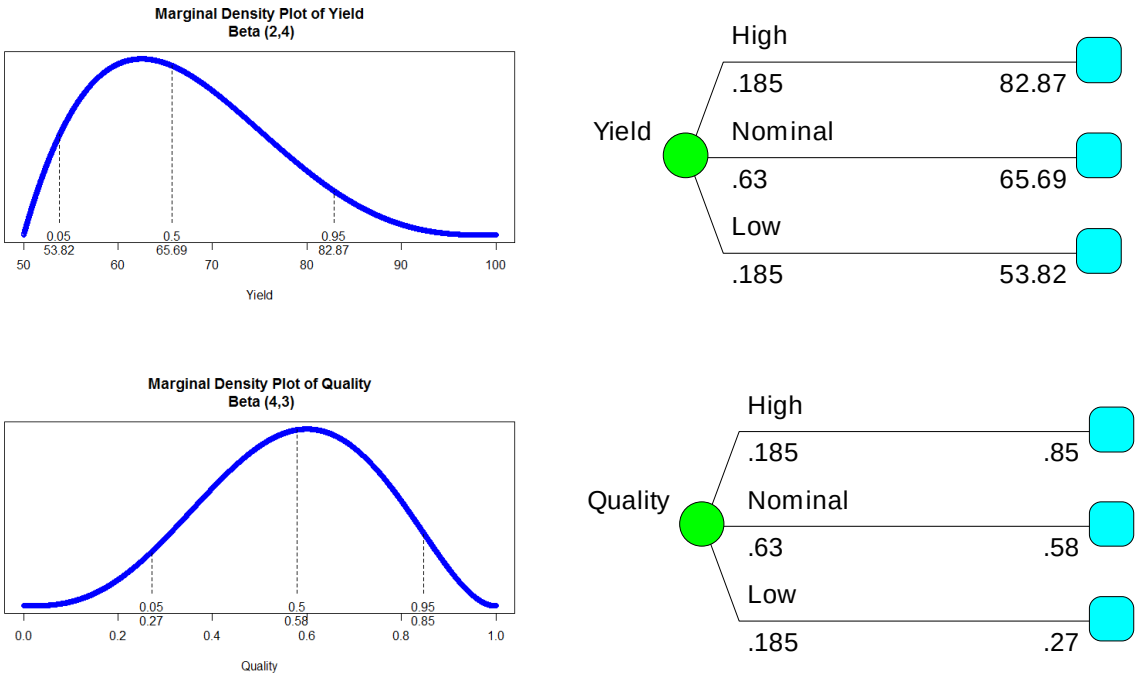


Figure 4: Discretization of Continuous Distribution

1.4.3 SOLUTION

Solving the decision tree using the DPL software, we obtain the optimal policy tree shown in Figure 5. It shows that the defender should implement the containment and surveillance safeguards but should decline to do inventory verification because of the latter's high cost relative to the decrease in probability of the attacker's success (bold paths in the decision tree indicate the optimal path for a decision maker; e.g. it is better for the defender to implement the inventory verification safeguard in Figure 5). The attacker chooses to attack based on these defender choices since the expected losses to the defender are larger for an attack compared to the no attack option.

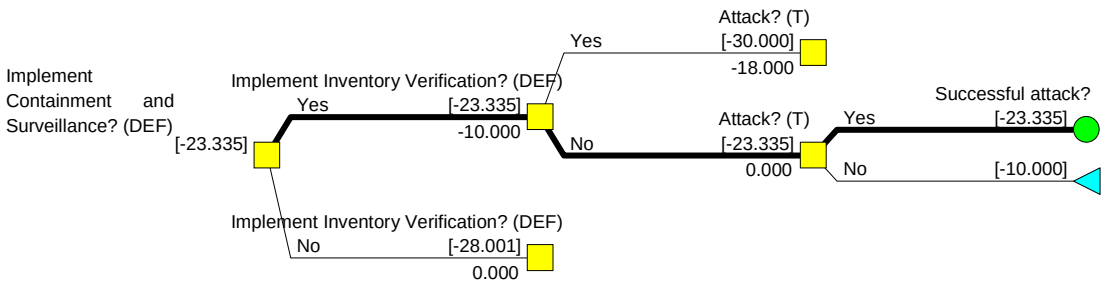


Figure 5: Policy Tree for the Simple Example

In Figure 5, the values in brackets represent the optimal value to the defender from that point forward in the tree. The following steps explain how the values are obtained. The expected value of the yield in Figure 4 is 0.185*82.87+0.63*65.69+0.185*53.82=66.673, which represents the expected value of a successful attack. If the defender's strategy is to implement containment and surveillance (C/S) but not inventory verification, then the probability of a successful diversion is 0.2 (refer to Table 1). This means the expected value of an attack is therefore 0.2*66.673=13.335. The defender incurs a cost of 10 to implement the safeguard resulting in an expected cost to the defender of 23.335. This value is captured in Figure 5 by the bracketed value at the leftmost edge of the tree. The non-bracketed values represent the costs incurred by that decision. The choice to implement C/S costs 10 and you can see in the upper most branch of the figure that implementing inventory verification costs 18. From each node in the tree, the optimal expected value is shown in brackets. If the defender chooses not to implement C/S then the best expected value he can obtain is 28.001

Our general decision tree framework for the analysis of a fuel cycle facility can be summarized as follows. The defender makes decisions about the safeguards to implement, considering the costs of implementation and the influence those decisions have on the probability of detection (and thus on the attacker's decisions). The attacker uses this information to make an optimal choice regarding whether or not to attack the target. Hence, the model provides a measure of the deterrent value of security measures.

The corresponding influence diagram shown in Figure 6 can also illustrate the relationships among the decisions, uncertainties, and parameters in this model. The defender's actions affect the probability of detection, which in turn influence the chance of success. Given a successful attack, the uncertainty over the quality of material stolen and the yield are characterized by probability distributions.

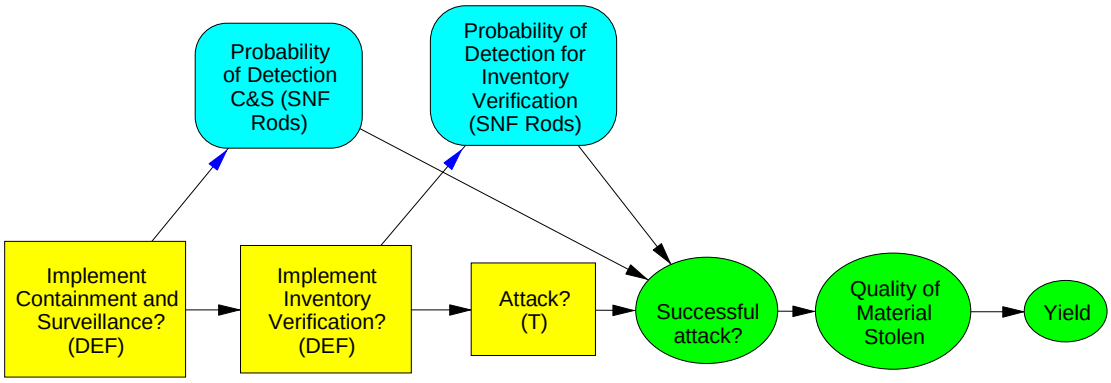


Figure 6: Influence Diagram for Small Example

In Section 2 we will examine a more comprehensive model with a larger set of defender and attacker strategies including additional targets in an UREX+ aqueous reprocessing facility. We also consider an example of protracted theft. The model will also be formulated as a mixed-integer program (MIP) to handle a larger scale problem and will be solved using GAMS optimization software.

2 STRATEGIES

2.1 DEFENDER-ATTACKER STRATEGY MATRIX

The example from Section 1 used a limited number of defender-attacker strategies. These were selected from a more comprehensive list related to an UREX+ aqueous reprocessing facility developed by Ward (2012) and based on work by Durst, et al. (2007), Pereira (2008), and Todd (2008).

Figure 7 from Ward (2012) depicts the process steps at a UREX+ aqueous reprocessing facility. Fuel is received in spent fuel bundles from a reactor. Fuel attributes are measured using non-destructive assay techniques (NDA) and operator declarations are compared to inspector burn-up calculations and NDA measurements. The fuel is then stored until use. The storage is under constant containment and surveillance (C/S), including cameras and directional radiation detectors.

The spent fuel enters front-end operations and is first mechanically chopped and sheared. The spent fuel pellets are then dissolved in nitric acid; undissolved material, including cladding and undissolved fuel, are removed from the process stream ("hulls"). The remaining material is known as raffinate, and it then enters a series of centrifugal contactors or mixer settlers that comprise the UREX extraction step. In this step, uranium and technetium are co-extracted from the solution using tributal phosphate (TBP) as a solvent. The uranium and technetium are then separated and stored independently.

After the uranium is extracted from solution, the remaining solution contains TRU products (Pu, Np, Am, Cm), fission products and lanthanides. CCD-PEG solvent is used to extract the major lanthanides from the solution, namely Cs and Sr, which pose a large repository burden due to their short half-lives and high heat generation rates. The solution is then sent to the TRUEX process, where fission products are extracted to be stabilized and stored. Finally in the TALSPEAK process phase, the TRU, including plutonium, is separated out of the solution. The plutonium is left with the TRU and is stored in solid form on-site until it is shipped.

In Figure 7, the red boxes show potential points of diversion in an aqueous UREX+ process. The possible points of diversion are: diverting spent fuel from storage, diverting TRU into the hulls, diverting material from any of the solvent extraction steps, or diverting TRU product from storage. Diversion of solution into hulls or from the extraction steps can be done with or without replacement with nitric acid to maintain mass and volume levels.

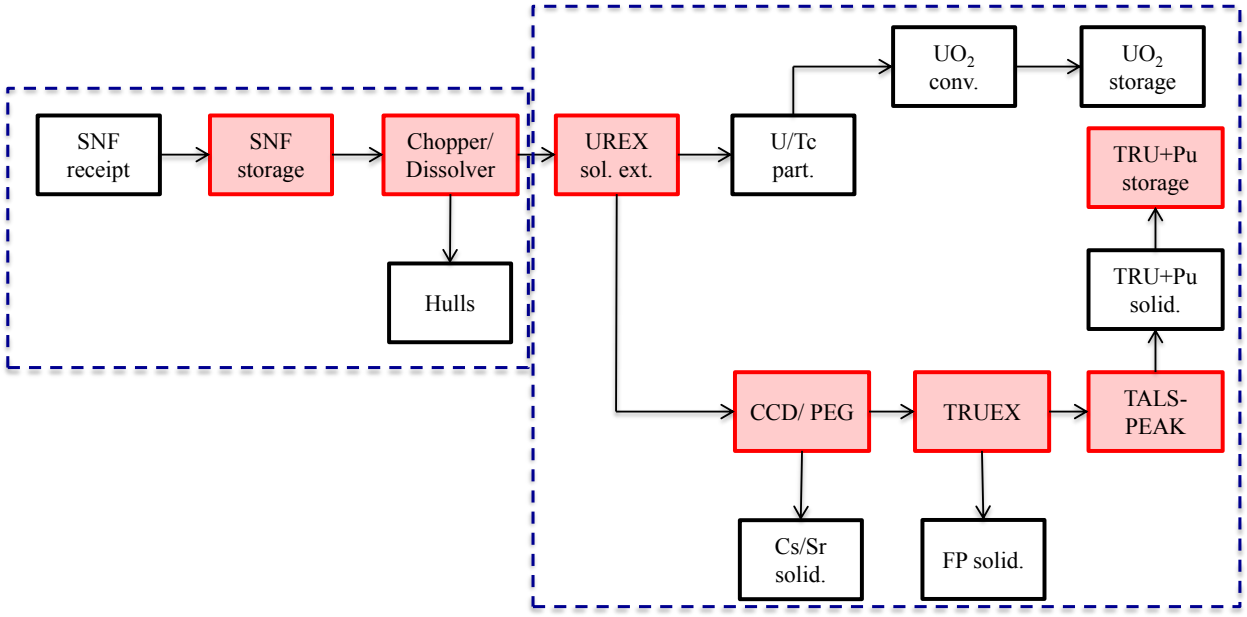


Figure 7: Diagram of a UREX+ aqueous reprocessing facility. Source: Ward (2012)

Within the context of an aqueous UREX+ reprocessing facility, Ward (2012) identified seven attacker options for the theft or the diversion of radioactive materials. In addition, she identified ten potential defender options to safeguard these materials at a domestic facility. Then, for each of the attacker/defender pairs, she indicated whether the defender strategy would be effective in detecting the attacker option in a matrix (see Table 2). The names of these strategies shown in Table 3 and additional details regarding the defender strategies are provided in Appendix A.

Table 2: Defender-Attacker Strategy Matrix

	Attacker Strategies						
Defender Strategy	**1**	**2**	**3**	**4**	**5**	**6**	**7**
A	X					X	
B	X	X	X	X	X	X	
C		X	X	X	X		
D			X		X		X
E			X		X		X
F	X	X	X	X	X	X	
G		X	X	X	X		
H		X	X	X	X		
I		X	X	X	X		
J							X

Source: Ward (2012)

An X in the defender-attacker strategy matrix indicates that the safeguard strategy in the leftmost column of that row is an effective countermeasure to the attacker strategy in the uppermost cell of that column. For example, the X in cell (F,2) means that safeguard F (containment and surveillance, C/S) can be deployed to mitigate the risk from attacker strategy 2 (Divert solution into hulls). The absence of a X in cell (F,7) indicates that safeguard F

(containment and surveillance) has no ability to effectively mitigate risk from attacker strategy 7 (Falsify spent fuel declarations).

Table 3: Strategy Descriptions

Attacker Strategies	Defender Safeguards
1. **Divert spent fuel rod** 2. **Divert solution into hulls** 3. **Divert solution into hulls with replacement** 4. **Divert solution from extraction** 5. **Divert solution from extraction with replacement** 6. **Divert TRU product from storage** 7. **Falsify spent fuel declarations**	A. Item Counting B. Inventory Verification (interim or annual) C. Design information verification D. Non-Destructive Assay techniques (Gross neutron counting; Pu/Cm-242 ratio counting) E. Destructive Analysis F. Containment and Surveillance (C/S) G. Solution Measurement and Monitoring System (SMMS) H. Plutonium Inventory Measuring System (PIMS) I. Hybrid k-edge densitometry J. Lead slowing-down spectroscopy

Source: Ward (2012)

The attacker strategies considered are primarily diversion/theft scenarios. Therefore, the defender strategies in Table 2 are restricted to actions that are relevant for diversion/theft attacks.

Defender strategies have varying levels of effectiveness depending on the attack scenario. Recall the example from Section 1. In Figure 3, we assumed that implementing containment and surveillance resulted in a probability of detection of 0.8. This was specific to that attack scenario. The probability of detection for a different attack scenario could be 0.9, for example.

3 DECISION TREE MODEL

We use a decision tree to represent a sequential leader-follower (Stackelberg) game theory formulation. An initial set of decisions is made by a defender regarding which safeguards to implement to mitigate the risk of a terrorist attack. This set of safeguards represents the defender's strategy in the game. After observing the defender's strategy, the attacker then decides whether or not to attack a set of potential targets. The positive-valued payoffs represent a loss to the defender and therefore the defender wishes to minimize the expected value of his payoff. The attacker desires the opposite, to inflict as much damage on the defender as possible. The attacker's objective is to maximize these expected payoffs.

3.1 STRATEGIES

We now consider a broader selection of targets and safeguards in our illustrative model than we did for the simple model of Section 1. However, incorporating all of the possible strategies from Table 2 would be intractable for the purposes of this report. A smaller subset was taken that demonstrates the complexity of the problem and the need for alternative formulations while still providing a credible model with realistic aspects of nuclear fuel cycle security. The reduced defender-attacker matrix used (Table 4 and Table 5) is adapted directly from Table 2 and Table 3 (Ward, 2012).

Table 4: Reduced Defender-Attacker Matrix (top), Strategy List (bottom)

	1	**3**	**4**	**6**
A	X			X
B	X	X	X	X
F	X	X	X	X

Defender Safeguards	**Attack Scenarios**
(A) Item Counting	(1) Divert Spent Fuel Rod
(B) Inventory Verification	(3) Divert Solution into Hulls (w. replacement)
(F) Containment and Surveillance	(4) Divert Extraction Solution - UREX
	(6) Divert TRU Product from Storage

Table 5: Strategy Descriptions

Identifier	**Strategy**	**Description**
A	Item Counting	Used to identify missing discrete items, e.g. spent fuel rods, solid TRU product ingots
B	Inventory Verification	Identifies diverted materials through monthly/interim and annual inspections
F	Containment and Surveillance	Includes cameras and other surveillance equipment. Helpful in storage areas where little movement is generally expected.
1	Divert spent fuel rods	Theft of spent fuel rods from spent fuel storage area
3	Divert solution into Hulls	Can be done with or without replacement. Without replacement can be detected by mass and volume measurements. With replacement could pass mass and volume measurements, but the solution density will be altered
4	Divert extraction solution from	Also can be done with or without replacement; same considerations as solution diversion into hulls. This includes options to steal from UREX

	Storage	solution extraction, CCD/PEG, TRUEX, or TALS-PEAK
6	Divert TRU product from storage	Diversion of TRU product from the TRU+Pu product storage area

Source: Rebecca Ward (2012) Defender-Attacker Matrix

As we discussed earlier, the defender's strategy comprises a set of safeguard decisions. The number of strategies is determined by calculating the number of combinations of individual safeguard options. In this illustration, the first two safeguards have two options (yes/no) and the last safeguard has four options (cameras, detectors, both, or none). This results in $2 \times 2 \times 4 = 16$ strategies which are fully enumerated for clarity in Table 6.

Table 6: Enumeration of defender strategies

Strategy	Item Counting	Inventory Verification	Cameras	Rad. Detectors	Both
1	Y	Y	Y		
2	Y	Y		Y	
3	Y	Y			Y
4	Y	Y			
5	Y		Y		
6	Y			Y	
7	Y				Y
8	Y				
9		Y	Y		
10		Y		Y	
11		Y			Y
12		Y			
13			Y		
14				Y	
15					Y
16					

We illustrate these strategies in the context of an aqueous UREX+ reprocessing facility (Ward 2012). These are realistic safeguards a defender can implement in the facility and could also be viewed as improvements to existing safeguards currently in place. The defender can choose any combination of safeguard options, resulting in a growing number of strategies (which includes the set of safeguard choices). For example, on strategy implements item counting and inventory verification, but no containment and surveillance safeguards. A different strategy (Strategy 1) would include the previous safeguards but add cameras as well.

Whereas the defender can choose a portfolio of options which is considered as one strategy, we limit the attacker to choosing one attack scenario, including the option to not attack. The exception to this is the case of a protracted theft which we discuss later.

3.2 Decision Tree

The commercial decision tree software DPL was used to model the strategies shown in Table 6. The collapsed full tree is provided in Figure 8 below. The defender's choices are made and then the attacker, having observed those

decisions, chooses a target to attack (if any). If the terrorist does attack a target and is successful, the attacker and defender then observe uncertain probability distributions for the quality of the material and the yield obtained.

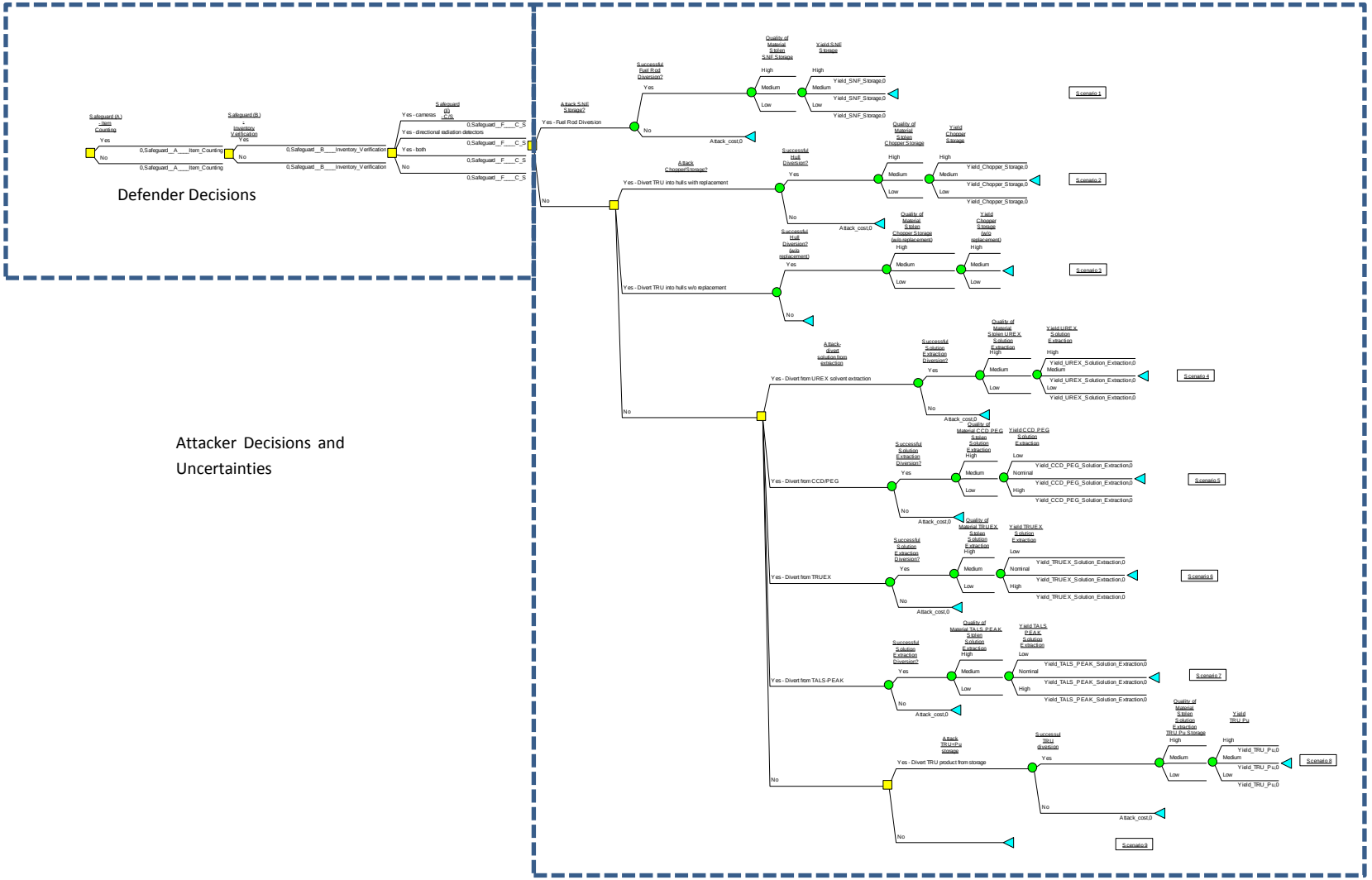


Figure 8: Decision Tree Formulation with 3 defender safeguards, 9 attack scenarios

As we have mentioned, the defender's decisions compose a particular strategy. Each path through the defender decision tree represents a particular strategy from Table 6 and is illustrated in Figure 9. Strategy 1 consists of implementing item counting, inventory verification, and a containment and surveillance system with cameras. Strategy 9 is similar but with no item counting safeguard. Strategy 16 is the choice to deploy none of the safeguards considered; again this may be more accurately described as the decision to not improve safeguards already in place. In Figure 8 we model each of the attacker's choices as a unique decision node, e.g. Yes Divert from Storage and Yes Divert from UREX. This is done for modeling convenience, and the full set of attacker decisions could be combined into a single decision node analogous to the representation in the simple model in Figure 2.

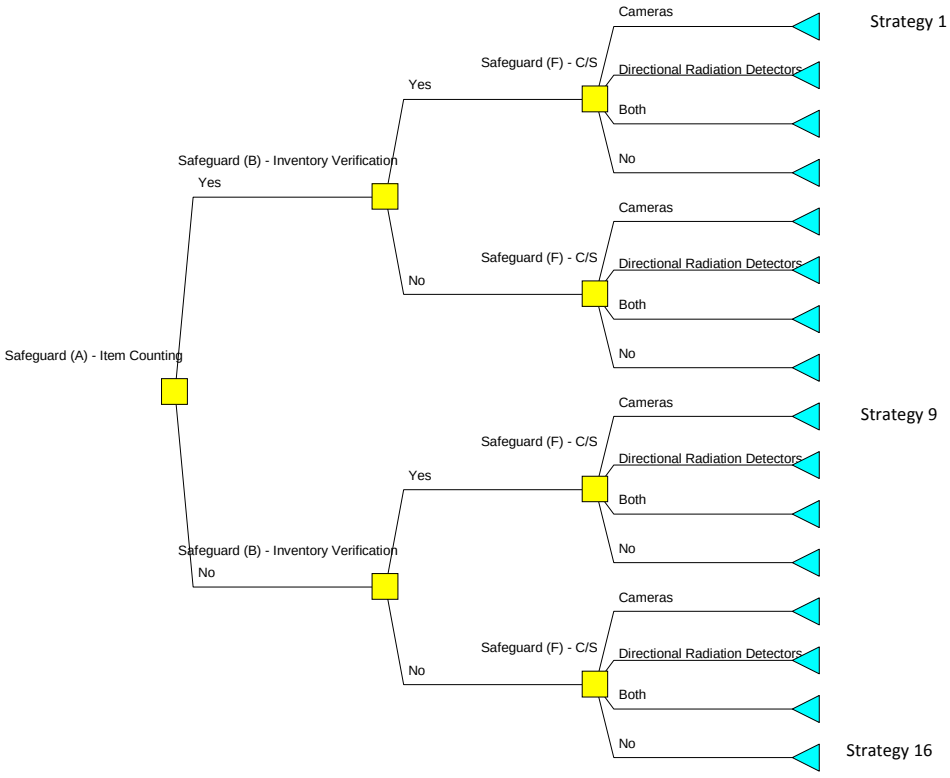


Figure 9: Defender's Strategy

It is important to recall that the safeguard selections influence the probabilities of a successful attack. The likelihood of a successful diversion of spent fuel rods, for example, depends on which safeguards are implemented. The right hand side of Figure 8 shows the full decision tree for the attacker, including the resulting uncertain payoffs. However, we should call attention to the fact that this subtree appears in the model 16 times, once for each defender strategy, and would be appended to each of the 16 end nodes in Figure 9.

The model in Figure 8 only reflects a subset of defender and attacker scenarios from Ward (2012); the fully expanded tree is much larger than our simple example. The decision tree representation is advantageous because of the visibility of the decision structure and the decisions of the defender and attacker. The policy tree generated by the DPL software is shown in Figure 10 and illustrates the ease with which the optimal strategies can be identified.

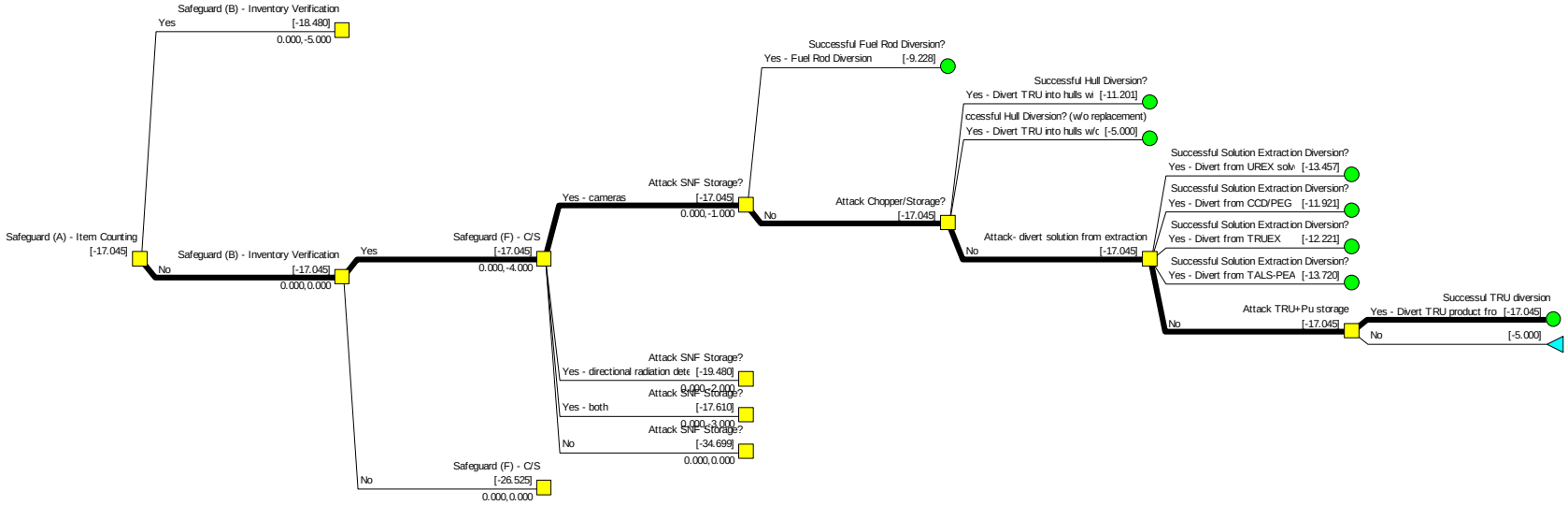


Figure 10: Policy Tree

The interpretation of the tree in Figure 10 is as follows. The defender chooses to implement inventory verification and cameras from his available options and incurs a cost of 5. The attacker, observing these choices, then determines their optimal attack strategy. Basing his decision on the expected value (including the payoffs and success uncertainties), he chooses to attack the TRU+Pu storage area to divert TRU product. As highlighted in Figure 11 (a zoomed in version of Figure 10) the attacker observes that the safeguards implemented leave a 14% chance of a successful attack and weighs that against the potential payoffs. The expected value of the attack, if successful, is 78.748 to the attacker. Considering the loss of 2 incurred for a failed attack, the attacker has an expected value for this scenario of 17.045. The values shown in the figure are calculated in the same fashion as those in Figure 5 and the interpretations of the bracketed and non-bracketed values are the same.

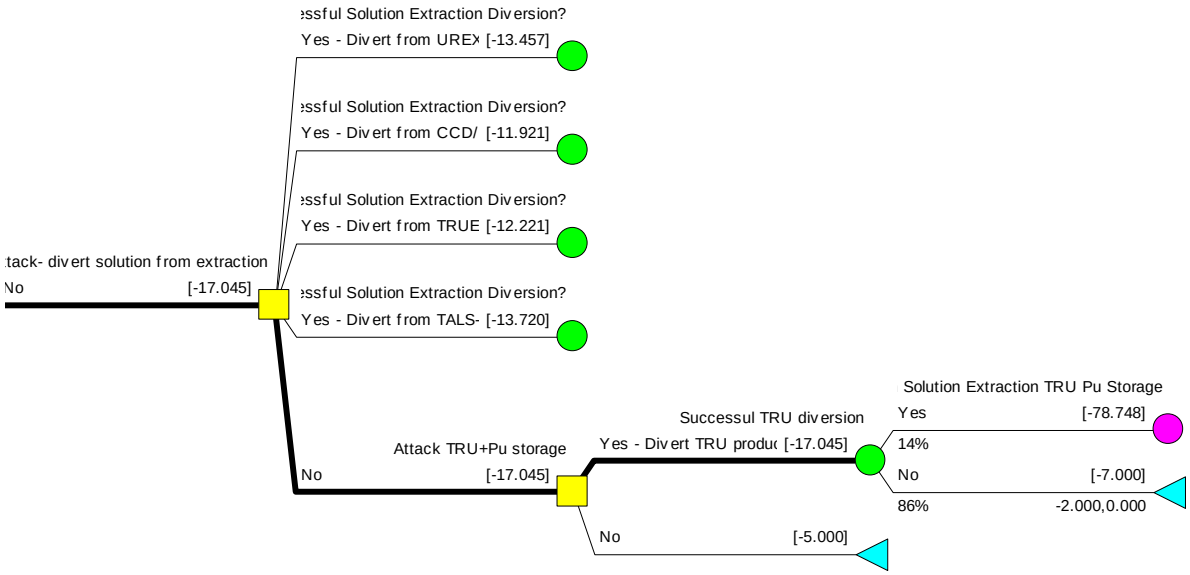


Figure 11: Subsection of Figure 10, showing the expected values

Notice that the scenarios to attack by solution extraction diversion (i.e. UREX, CCD/PEG, TRUEX, TALS-PEAK) yielded expected values ranging from 11.921 to 13.720, which are less desirable to the attacker than the 17.045 for the chosen optimal strategy. Had the defender chosen a different strategy, we would be looking at a different section of the tree with different expected payoffs because of the different probabilities of successful diversion.

The decision tree offers an easy way to view and interpret the decisions made. However, it becomes unwieldy once a larger set of defender and attacker options are included, and necessitates reformulation as a mathematical program. Figure 8, while showing the entire tree structure, does not show the full set of branches and endpoints. To illustrate, consider how the policy tree looks when only the post-defender choices are expanded fully in Figure 12.

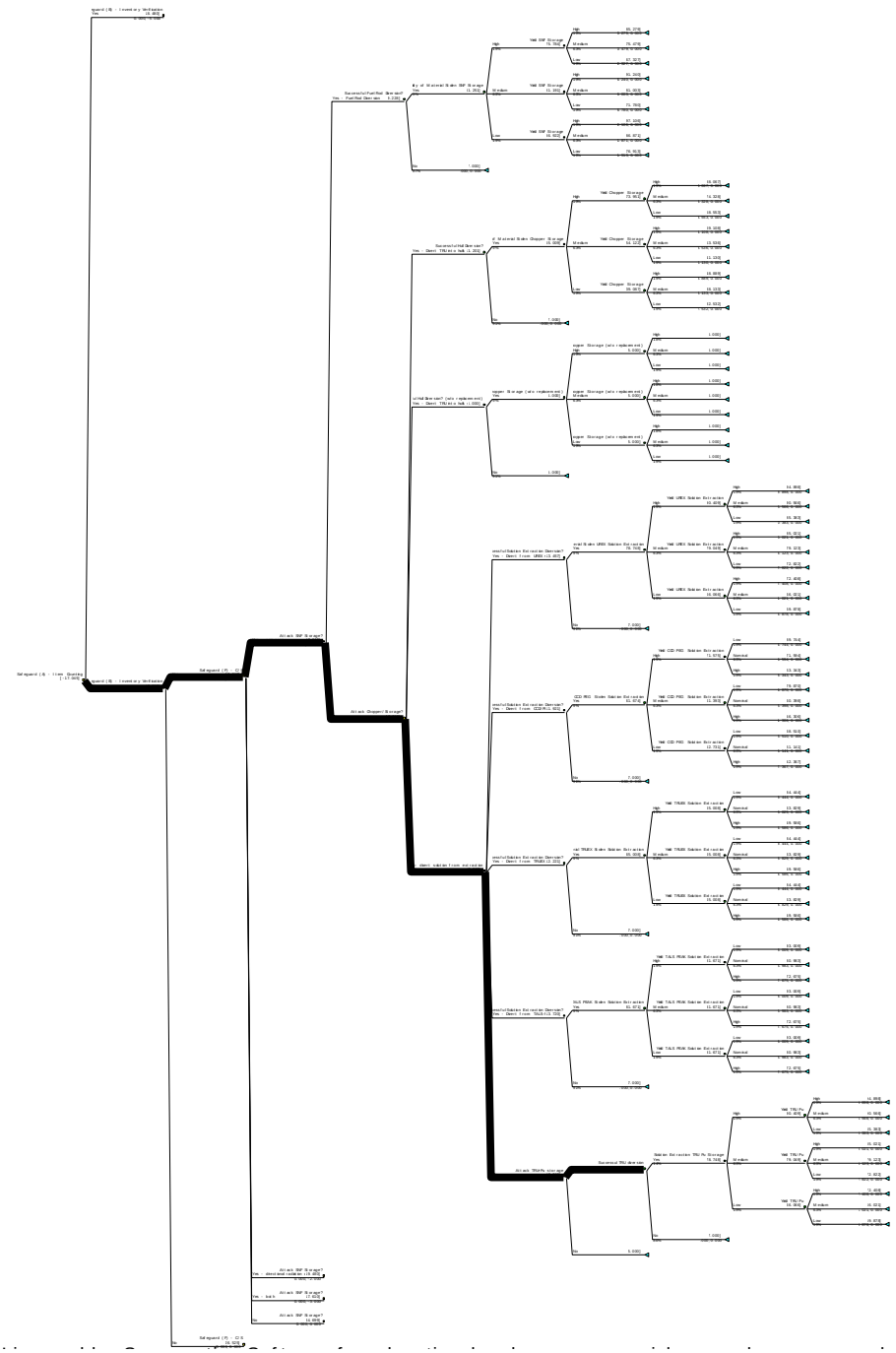


Figure 12: Expanded policy tree

3.3 UNCERTAINTIES

As shown in the previous examples, the attacker considers the expected loss to the defender for each attack scenario. The probability of a successful attack depends on the safeguards implemented by the defender. The probability distributions of the Quality of Material Stolen and the Yield are assessed as continuous beta distributions. The decision tree uses the discretized equivalents of these probability distributions obtained by the Extended Pearson-Tukey method. Section 4 provides the background on this methodology.

3.4 Probability of a Successful Diversion or Theft

An attack is successful only if the attacker is undetected by all safeguards used by the defender. For example, if the attacker was undetected by the item counting safeguard but was caught by radiation detectors, then we assume the attack was a failure. We also treat the safeguards as independent of each other in the sense that the probability of passing one safeguard is independent of passing another. These assumptions can be combined to state that the probability of attack 'a' being successful is shown below where P_{da} is the probability that attack 'a' will be successful if the defender chooses strategy 'd'. P_{sa} is the probability of safeguard 's' detecting attack 'a.'

$$P_{da} = \prod_{s \in S} (1 - P_{sa})$$

Furthermore, not all safeguards are useful in detecting all types of threats. In the attacker-defender matrix of Section 3.1, we saw that Safeguard A (item counting) detects only a subset of the attack scenarios (those indicated with an X in the attack column of Table 4). We consider the proposed safeguards in this model as improvements to existing security measures. Therefore, it is reasonable to assume that a safeguard not included in the defender's strategy does not alter the probability of a successful attack. In this case, the contribution of the unselected safeguard to the probability of detection would be 0, so there would be no effect on the probability of a successful diversion. For example, assume there are two illustrative safeguards and safeguard 1 is implemented with detection probability 0.9, while safeguard 2 is not implemented. The probability of a successful diversion in this case is $P_{diversion} = 1 - .9 * 1 - 0 = 0.1 * 1 = 0.1$.

3.5 Protracted Theft

A protracted theft by the attacker is one in which an attack comprises a series of repeated thefts from one target. This is done in order to steal small amounts of special nuclear material (SNM) that can be cumulatively used to build a weapon or other device. This type of attack scenario can be considered within our framework as illustrated with a simple hypothetical example with one defender safeguard and one attacker target. This is shown in Figure 13 and discussed in the remainder of this section. As is the case with all probabilities and payoffs in this report, the values provided are notional and used only to illustrate the logic. They are not drawn from any government sources.

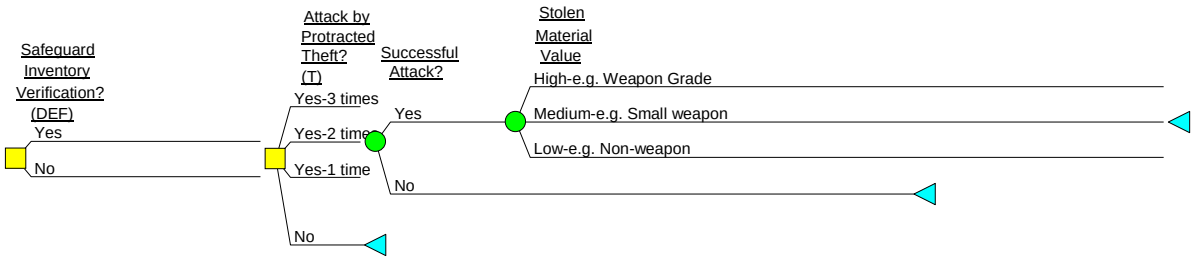


Figure 13: Protracted theft decision tree

In this example, the attacker is considering an attack by stealing special nuclear material (SNM) over one, two, or three stages, or not at all. To protect against such an attack, the defender can deploy a safeguard to detect SNM thefts. Safeguards that can detect thefts or diversions are material accountability systems which, among other things, can account for inventory levels and also detect changes in the mass, volume, and density of fluids (Ward 2012). These accounting systems are subject to measurement error, and protracted thefts are aimed at stealing

quantities small enough to fall within the normal measurement error. For example, if a one-time theft lowered the mass of some SNM storage level but was still within the error tolerance, a review of the measurements would not necessarily indicate a theft. However, if a series of thefts occurred, the resulting sequence of low (but within error tolerance) measurements could be cause for alarm. Figure 14 provides an illustration of this idea.

The dashed blue line in Figure 14 plots a hypothetical set of readings for an inventory verification system that measures the mass of some fictional SNM in storage. Readings below the low threshold or above the high threshold would be immediate cause for concern. As a result, any theft over multiple time periods or stages needs to be small enough as to stay within these bounds. The dashed blue line shows a sequence of measurement readings that should give no cause for concern to the defender because it appears to be random and unbiased. The dotted orange line in Figure 14 reveals a set of readings that are very low in sequence but still within the threshold. The randomness shown in the first plot does not fully disguise the steady decrease from time 3 to 6, and this trend, although within the threshold levels, should be an indication to the defender that there *may* have been a theft during that period. The conclusion here is that a sequence of decreasing readings within the threshold should make the defender more vigilant in monitoring the area to increase the chance of detecting a theft.

In order to increase the likelihood of a damaging outcome to the defender, the attacker wants to steal as much material as possible. If the attacker is caught on any attempt, the attack is unsuccessful and results in a loss to the attacker. Therefore, the attacker wants to maximize the amount of SNM stolen but only if the complete attack (over one or more stages possibly) goes undetected by the defender.

To illustrate this logic using notional probabilities, assume there is a 0.70 chance that a particular safeguard would detect a one-time theft of SNM. If successful, the theft was undetected, and given our previous discussion, the subsequent measurement will be within the error bounds and there is no reason for the defender to be alarmed. Therefore, the probability of detection for a second, repeated theft would remain unchanged at 0.70. However, after two thefts, the defender could notice decreasing (but allowable) measurement readings and would now have cause for concern. This would make the defender more cautious and his increased vigilance would cause the probability of detection to increase to a number larger than 0.70 for subsequent diversions.

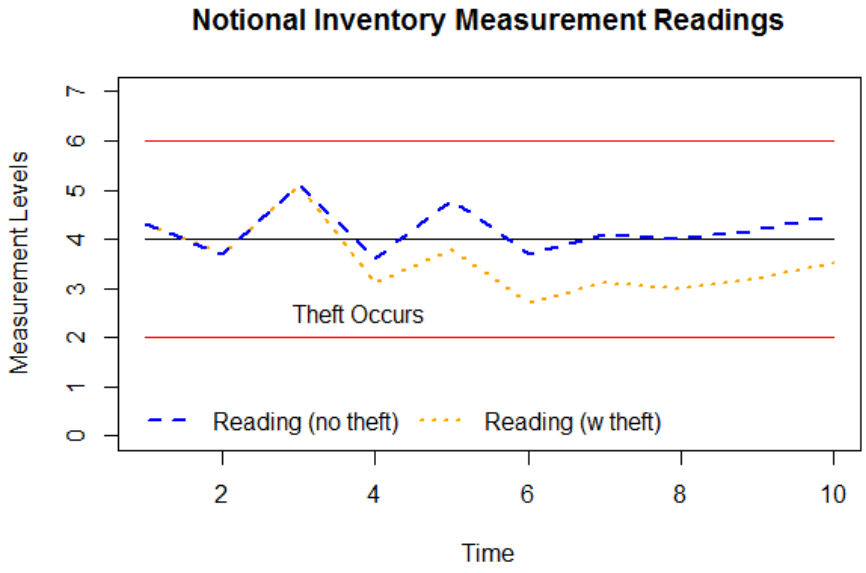


Figure 14: Fictional inventory measurement readings for protracted theft case

From the perspective of the attacker, the probability of being successful once is 0.3. To successfully steal SNM twice, the attacker must defeat the safeguard twice with a probability $0.3 * 0.3 = 0.09$. To successfully steal SNM three times an upper bound on the probability of detection should be $0.3^3 = 0.027$, or 2.7%. This is an upper bound because, based on our reasoning above, the defender may become more cautious after two consecutive decreasing measurements. For this example, we choose the probability of detection for a protracted theft over three attempts to be 0.99 and therefore the probability of a successful attack is 1% (which is less than the 2.7% upper bound). We do not prescribe an exact formula here as a function of n thefts, as these probabilities might be individually assessed by subject matter experts familiar with protracted theft and the defender's detection posturing following suspicious measurement readings from the safeguard. The DPL representation of the probability of detection, conditioned on the defender and attacker choices, is shown in Figure 15.

When the attacker considers his strategy choices, we assume that the attacker factors in the potential yield of the stolen quantity when considering how many thefts to attempt. A single theft would result in smaller quantities stolen and would more likely be useful for a smaller weapon or device with less impact. In this fictional example, larger quantities stolen as a result of multiple thefts could be used in a more damaging IND. To account for this, the probability distribution of the stolen material value is conditional on the number of thefts as shown in Figure 16.

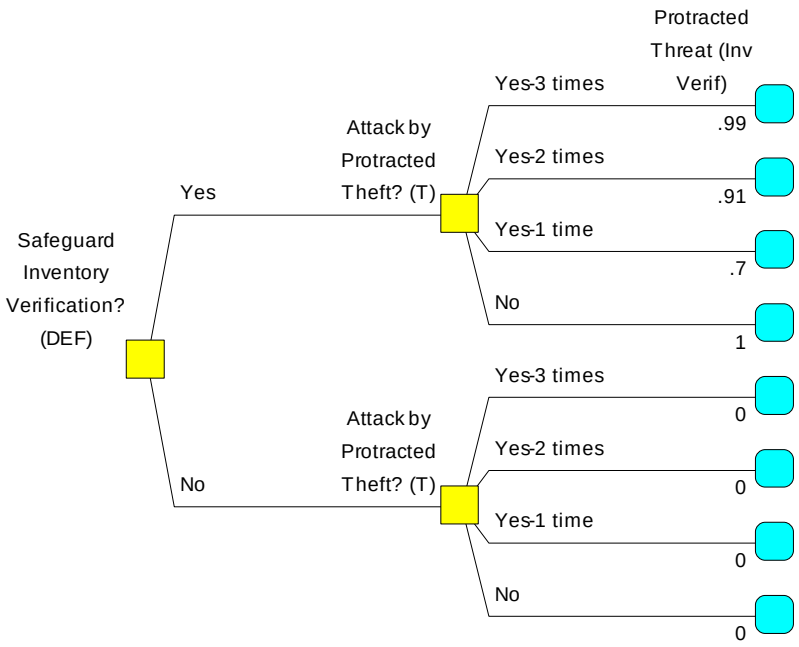


Figure 15: Conditional Probability of Detection

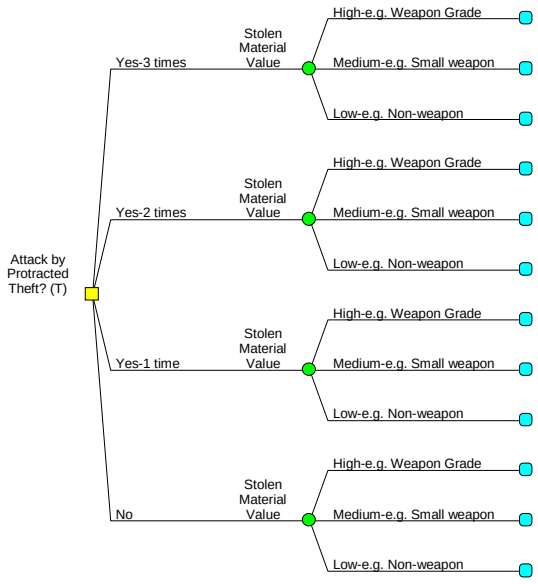


Figure 16: Protracted Theft Conditional Payoffs

Another assumption is that there is significant advance planning by the attacker. Since the type of device built is contingent on the amount stolen, it is reasonable to believe that the attacker makes a strategic decision on the amount they hope to steal, since the resources needed to steal, hide, and weaponize special nuclear material (SNM) could vary depending on the quantity of material. It would not make sense for the attacker to plan for building a large-scale weapon from a very small amount of SNM. Similarly, it would not be sensible for the terrorist to plan on stealing a significant quantity of SNM over multiple attacks only to build a small-scale weapon. In other

words, the protracted theft decision is made with a cumulative goal in mind. Therefore, we model the attacker's decision regarding the number of attacks as one initial decision of 0, 1, 2, or 3 attacks. We limit the number of thefts to 3 for this example, but a realistic example could include more if experts wish to model this.

The DPL software representation of the optimal decision given the arbitrary and notional probability and payoff parameters is illustrated below. The defender's objective is to minimize the expected payoff, as these positive payoffs are actually losses to the defender. The attacker aims to maximize the expected value. The game and notional payoffs are shown in Figure 17.

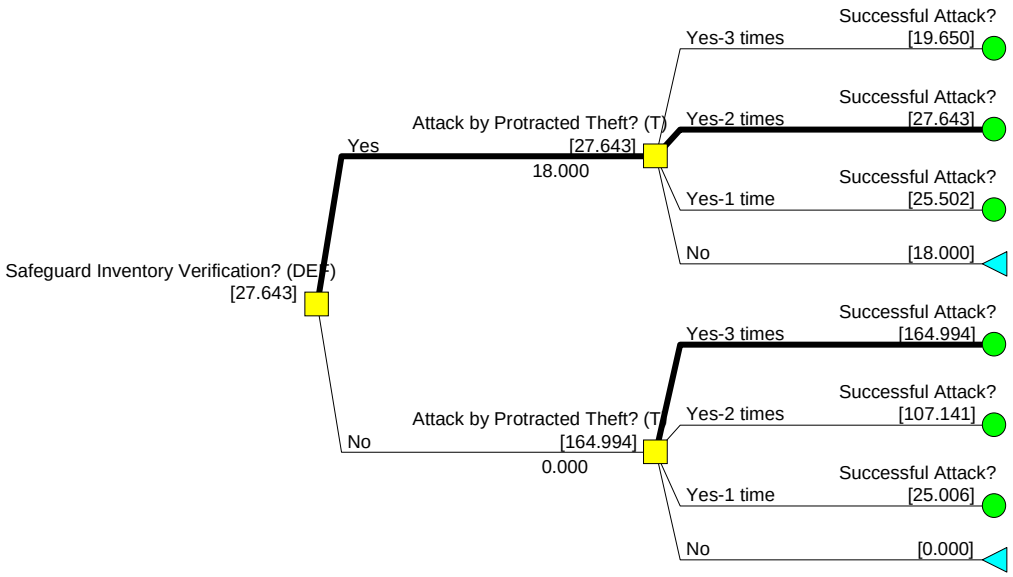


Figure 17: Protracted Theft Policy Tree

As shown in the lower half of Figure 17, if the defender does not implement inventory verification, an attacker theft over three attempts yields the highest expected value (164.994). If the defender does implement the inventory verification system, then the attacker chooses to steal over two attempts, and the expected payoff is much smaller now, only 14.643 (driven largely by the smaller probability of success with the safeguard present). The other expected payoffs are also comparatively smaller, ranging from 5 to 12.502. Facing expected payoffs of 14.643 and 164.994, the defender chooses to implement the safeguard to minimize the risk faced by a protracted theft attack.

The values come from the notional beta distributions as shown in

Table 7. The Extended Pearson-Tukey distribution was used to discretize each distribution. For example, a successful protracted theft over 2 attempts results in discrete payoffs of 127.60, 107.49, and 85.48 with probabilities 0.185, 0.63, and 0.185, respectively. This results in an expected payoff of 107.141. However, the probability of a successful diversion, if the safeguard is implemented by the defender, is 1-0.91=0.09 from Figure 15. Therefore, the expected payoff for a 2-phase diversion attempt is 9.643. Assuming that implementing the safeguard costs 18, the expected value to the defender is 27.643.

Table 7: Distributions for Protracted Theft Example

Distribution	alpha	beta	lower bound	upper bound
Protracted Theft (3 times)	15	5	60	200
Protracted Theft (2 times)	8	6	50	150
Protracted Theft (1 times)	3	15	10	100

This hypothetical example illustrates how protracted theft can also be included in the decision tree as a single attack scenario with mutually exclusive options. The payoffs could also be adjusted to account for the correlation between Quality of Material Stolen and Yield payoffs as previously demonstrated. The structure used here was merely to illustrate the logic for this type of attack scenario.

4 CORRELATED UNCERTAINTIES

4.1 BACKGROUND

Modeling uncertainty is a crucial part of decision and risk analysis. In reality, multiple sources of uncertainty commonly exist, and they are likely to be correlated. Decision and event trees have become fundamental tools for modeling uncertainties due to their visual representation and ability to communicate complex relationships. However it is computationally difficult to incorporate dependencies among uncertainties in a tree, particularly given the practice of utilizing discrete probability distributions.

When dealing with continuous uncertainties, it is common to discretize the probability density functions into discrete probability mass functions. This reduces the solution space, eases assessment, and can aid in the comprehension of the tree for the decision makers. For a recent review of the literature on the best discretization schemes see Hammond and Bickel (2012). In this report we rely on the extended Pearson-Tukey (EPT) approximation that has been found to be a robust approach (e.g. Keefer and Bodily, 1983). The EPT uses probabilities of 0.185, 0.63, 0.185 for the 5^{th}, 50^{th}, and 95^{th} percentiles, respectively, of the target continuous distributions. Refer to Figure 4 for an illustration.

A natural approach to modeling multivariate uncertainties in a tree is to specify the conditional distribution of each uncertainty given the discrete outcomes of the preceding uncertainty(s). Despite its simple logic, this approach is limited in practical applications because conditional distributions are not easy to derive analytically, and the number of required conditional probability assessments increases combinatorially with the number of uncertainties; therefore, this approach is information-intensive and may be impractical for subjective risk assessment.

Wang and Dyer (2012) develop a practical approach for capturing dependencies in a tree that requires only the marginal distributions and measures of pair-wise correlations among the uncertainties. It also allows arbitrary marginal distributions. This procedure offers computational advantages that make it much more practical to implement for modeling correlated trees. For ease of exposition, our example focuses on a setting with only two uncertainties, but the reader is referred to Wang and Dyer (2012) where the approach is demonstrated for an arbitrary number of uncertainties.

Continuing with our example from Section 2, we consider two uncertainties, Quality of Material Stolen (Q) and Yield (Y), which may be correlated. If $F_q(Q)$ and $F_y(Y)$ represent the cumulative density function (CDF) of each uncertainty, respectively, then the joint CDF of Q and Y can be expressed as a copula function C,

$$F(Q,Y) = C(F_q(Q), F_y(Y))$$

The use of the copula C allows the separation of the marginal CDFs from the dependence structure, and the joint CDF $F(Q,Y)$ can be reconstructed from C, $F_q(Q)$, and $F_y(Y)$. Clemen and Riley (1999) used the multivariate normal copula to capture the dependence structure among random variables. For the two uncertainties case, a multivariate normal copula C_N is given by

$$C_N(F_q(Q), F_y(Y)) = \Phi_\rho (\Phi^{-1}(F_q(Q)), \Phi^{-1}(F_y(Y))) \quad (1)$$

where Φ_ρ is the cumulative distribution function for a standard bivariate normal distribution function with mean zero and correlation ρ, and the marginal cumulative distributions of Q and Y are transformed by the inverse of the standard normal distribution function Φ.

The most attractive features of the multivariate normal copula are its flexibility and analytical tractability. Combining the multivariate normal copula with marginal distributions, a large variety of multivariate distributions can be produced using the same generic procedure (Clemen and Reilly, 1999; Avramidis, Channouf and L'Ecuyer 2009). We restrict our attention to the normal copula because its properties fit the needs of the dependency-modeling problem. For an introduction to the theory of copulas and the discussions of different copulas that might be used in a tree, the reader may refer to Wang and Dyer (2012).

With the multivariate normal copula (Equation 1) modeling the dependencies among the random variables, we can transform a bivariate normal random vector Z with correlation coefficient r into the desired random vector (Q, Y). If we define $Z = (Q', Y') \sim N(Q', Y', r)$ then

$$(Q, Y) = (F_q^{-1}(\Phi(Q')), F_y^{-1}(\Phi(Y'))) \qquad (2)$$

where $F_a^{-1}(u) = \inf\{x: F_a^{-1}(x) \geq u\}$ for $0 \leq u \leq 1$, is the quantile function of the marginal distribution for uncertainty a with CDF $F_a(A)$.

Equation (2) is known as the NORTA approach in the simulation literature and was developed by Cario and Nelson (1997) as a simulation algorithm to generate multivariate random variables. Wang and Dyer (2012) apply NORTA to develop a four step procedure for modeling dependencies in trees. In what follows, we assume that the marginal CDFs, $F_q(Q)$ and $F_y(Y)$, are continuous for ease of exposition. We begin with the general process in Section 4.2 and then provide a numerical example of this procedure to discrete marginal distributions in Section 4.3.

4.2 Process for Continuous Uncertainties

Step 1. Assessment of Marginals and Correlation: The first step is the assessment of the continuous marginal distributions $F_q(Q)$ and $F_y(Y)$ and the measurement of r, the correlation Q and Y. We can assess any of the common correlation measures, the Spearman or Kendall's rank order correlations or the Pearson product moment correlations as the input correlation structure (Clemen and Reilly 1999; Clemen, Fisher and Winkler 2000; Cario and Nelson 1997). The decision maker can choose the appropriate correlation measurement based on the specific application and other considerations.

***Step 2. Specification of the Correlation Matrix Σ for Z*:** The second step is to calculate the corresponding Pearson product moment correlation r^* for r, the correlation between the original uncertainties Q and Y. We distinguish the correlation for the normal copula with superscript *, and the specified correlations for the original random variables with no superscript. For a specified Spearman's rank order correlation ρ_{qy} , $r^* = 2sin(\pi\rho_{qy}/6)$. For a specified Kendall's rank order correlation τ_{qy}, the formula is $r^* = 2sin(\pi\tau_{qy}/2)$ (Kruskal 1958). For a detailed discussion of correlation assessment methods see Clemen and Riley (2000), and Clemen, Fisher and Winkler (2000).

Step 3. Construction of the Base Bivariate Standard Normal Decision Tree: The multivariate normal distribution is one of the special cases that allow the conditional distributions to be expressed easily in terms of the marginal distributions and the correlation matrix. In this section, we will present a construction method for the multivariate standard normal decision tree for the standard bivariate normal vector Z by calculating the parameters of the conditional distribution.

Recall that Q' and Y' are two standard normal variables with marginals $Q' \sim N(0,1)$ and $Y' \sim N(0,1)$, respectively, and with Pearson product moment correlation r. A property of the bivariate normal distribution is that the conditional distribution of Y' given $Q' = q'$ is also normal, with conditional mean $\mu(Y'|q')$ and conditional variance

$\sigma^2(Y'|q')$.

Thus,

$$(Y' \mid Q' = q') \sim N(\mu(Y'|q'), \sigma^2(Y'|q'))$$

where $\mu(Y'|q') = r\,q'$ and $\sigma^2(Y'|q') = 1 - r^2$

Step 4. Point-to-Point Inverse Transformation: After the construction of the base bivariate standard normal decision tree in Step 3, we need to transform the standard normal representation to obtain correlated pairs from the desired marginal distributions. The standard normal cumulative distribution function is applied to each realization of the discrete approximation of the correlated standard normal variables, Q' and Y'; then Q and $Y \mid Q$ are obtained by applying the inverse of the target marginal distribution function for Q' and $Y' \mid Q'$, $F_q^{-1}(\Phi(Q'))$ and $F_y^{-1}(\Phi(Y' \mid Q'))$. This component-wise inverse marginal transformation procedure ensures that the resulting dependent marginal distributions are the target marginals as specified because, for example, $Q = q$ shares the same percentile as the discrete approximation of normal variable $Q' = q'$, and so $P(Q' = q') = P(Q = q)$; a similar argument holds for $P(Y'|Q' = q')$.

We now demonstrate how to apply this four-step process to our example of Quality of Material Stolen (Q) and Yield (Y).

Step 1: Assume that we assess the marginal distributions for our uncertainties as $Q \sim N(10,3)$, $Y \sim$ Beta(2,5) and Spearman correlation of $r = 0.5$, e.g. when an attacker is able to acquire material of above average quality, the Yield is expected to be above average, and vice-versa. The marginal distributions for these uncertainties are shown in Figure 18.

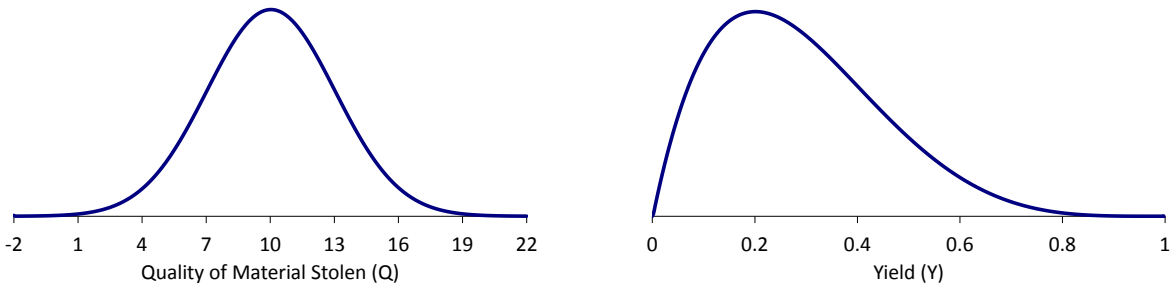


Figure 18: Notional Marginal Distributions of Uncertainties Q and Y

Step 2: To determine the correlation for the base standard normal tree we need to transform the assumed Spearman correlation as described above: $r^* = 2sin(\pi\rho_{qy}/6) = 2sin(\pi\,(0.5)/6) = 0.5176$.

Step 3: The discrete approximation for the standard bivariate distribution for the uncorrelated case, $r = 0$, is shown in Figure 19. For independent standard normal distributions the 5th (95th) percentiles are -1.64 and 1.64, respectively. Comparing the correlated standard bivariate distribution in Figure 19 ($r = 0.5 \rightarrow r^* = 0.5176$) to Figure 20, it is clear that the desired effect has been achieved: higher values of Q' are associated with higher values of Y', and vice-versa. For example, the 95th percentile of $Y'|Q'$ = High is 1.64 when there is no correlation (top branch of Figure 19) while the same quantity is 2.26 when including the assumed correlation (top branch of Figure 20).

Step 4: For ease of exposition we break Step 4 into two sub-processes. First, the cumulative standard normal probabilities of each fractile in the standard normal bivariate tree are calculated, i.e. $\Phi(Q')$ and $\Phi(Y' \mid Q')$, as shown in Figure 21 and Figure 22 (Figure 21 is a repeat Figure 22 for ease of comparison with Figure 23). For example, $\Phi(Q' = 1.64) = 0.95$ and $\Phi(Y' = 2.26 \mid Q' = 1.64) = 0.99$ in the top branches of Figure 21 and Figure 22.

The second sub-process of Step 4 is to evaluate the inverse of the CDF of the target marginal distributions, $F_q^{-1}(\Phi(Q'))$ and $F_y^{-1}(\Phi(Y' \mid Q'))$. For example, comparing the top branches in Figure 22 and Figure 23, 14.93 is the 95th percentile of Q ~ N(10,3) and 0.69 is the 95th percentile of $Y \mid Q = 14.93$ when Y ~ Beta(2,5). After the transformation to the desired marginal distributions, the effect of the positive correlation in Figure 22 persists in Figure 23: in Figure 23 when Q = 14.23 the 5th, 50th and 95th percentiles are 0.69, 0.42 and 0.18 compared to 0.37, 0.24 and 0.03, respectively, when Q = 5.07.

The discrete distributions in Figure 23 can be used in an event or decision tree to capture the marginal distributions of Q and Y and their desired correlation. For details on how to expand the four-step approach to an arbitrary number of marginal distributions and their association structure, see Wang and Dyer (2012).

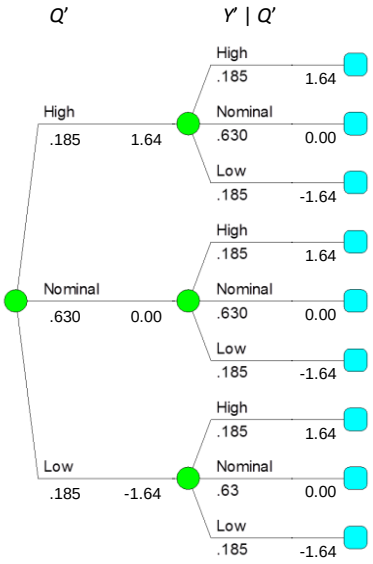


Figure 19: Bivarate Standard Normal Tree
$r = 0.0$, $Q' \sim N(0,1)$, $Y' \sim N(0,1)$

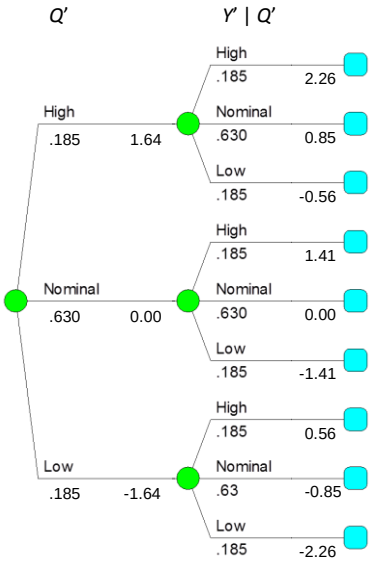


Figure 20: Bivarate Standard Normal Tree r=0.5
$r = 0.5$, $Q' \sim N(0,1)$, $Y' \sim N(0,1)$

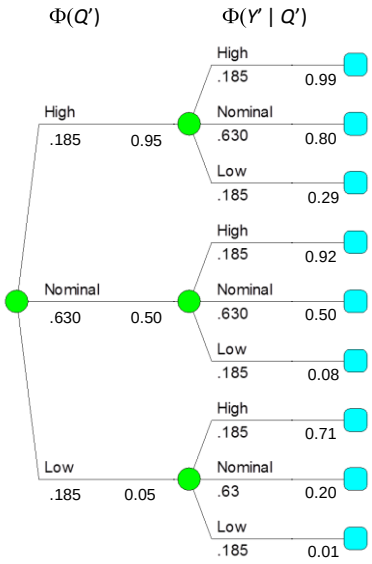


Figure 21: CDF of Binary Standard Normal Tree r=0.5
$r = 0.5$, $Q' \sim N(0,1)$, $Y' \sim N(0,1)$

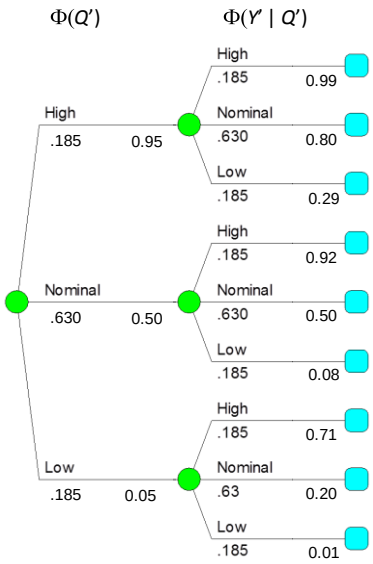


Figure 22: CDF of Binary Standard Normal Tree r=0.5
$r = 0.5$, $Q' \sim N(0,1)$, $Y' \sim N(0,1)$

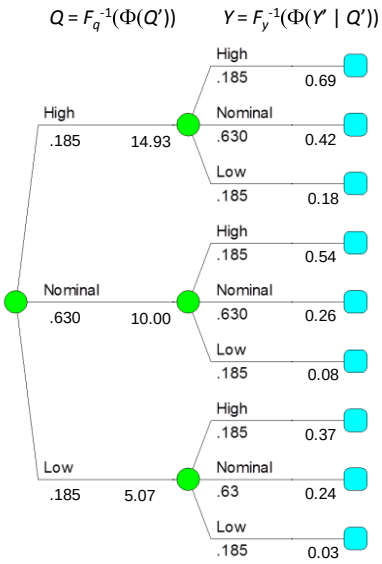


Figure 23: Event Tree for Marginals of Q and Y r=0.5
$r = 0.5$, $Q \sim N(10,3)$, $Y \sim Beta(2,5)$

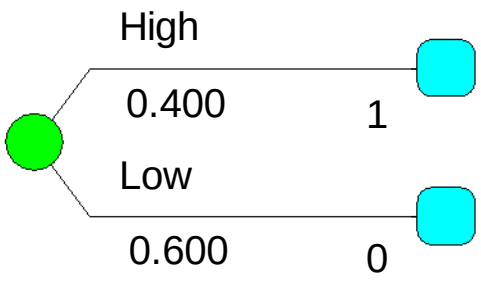
4.3 Process for Binary Uncertainties

In some cases the underlying uncertainties in a tree are naturally binary; e.g., Does Candidate X win the election? This is particularly common in pure event trees used in probabilistic risk analysis (PRA). Continuing with our example of the Quality of Stolen Material (Q) and Yield (Y) we redefine the variables such that they are binary or binomial as shown in the upper panel of Figure 24. We have arbitrarily assigned one of the outcomes to have a value of 1 that occurs with probabilities p_Q and p_Y, respectively, e.g., P(Yield = Success) = P(Y = 1) = p_Y. Using r, the correlation between Q and Y, we can derive closed form solutions for

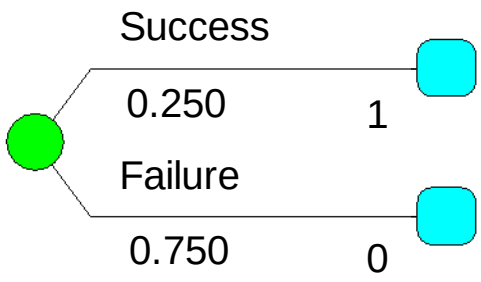
$$P(Y = 1 \mid Q = 1) = ((1- p_Q)/ p_Q)^{1/2} \times (p_Y - p_Y^2)^{1/2} \times r + p_Y \quad (3)$$

$$P(Y = 0 \mid Q = 1) = p_Y - (p_Q /(1- p_Q))^{1/2} \times (p_Y - p_Y^2)^{1/2} \times r \quad (4)$$

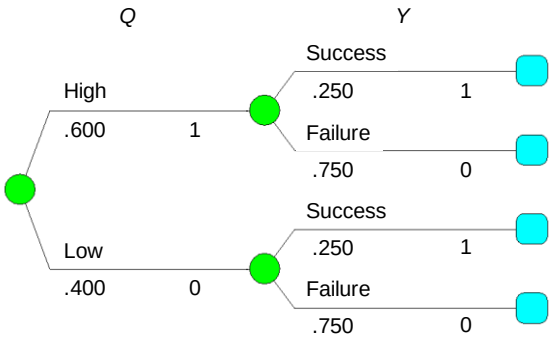
In the lower panel of Figure 24 we show an example of the use of equations (3) and (4) to generate an uncorrelated event tree (r = 0) and a correlated event tree when r = 0.5. The probability of successful yield is highest for correlated distributions and high quality material (0.515); which is higher than for uncorrelated distributions and high or low quality material (0.25); which in turn is higher than for correlated distributions and low quality material (0.073).



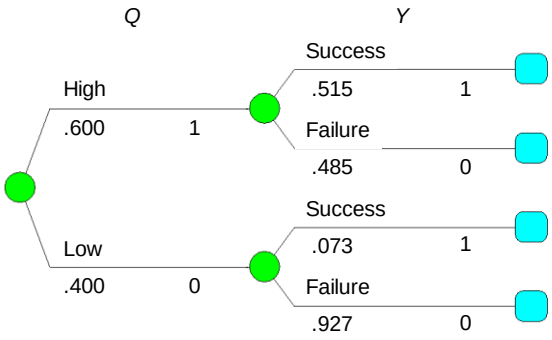
Marginal Distribution of *Q*



Marginal Distribution of *Y*



Marginal Distributions of *Q*, and *Y*|*Q* with *r* = 0



Marginal Distributions of *Q*, and *Y*|*Q* with *r* = 0.5

Figure 24: Discrete Marginal Distributions for Q and Y and Correlated Trees

As previously discussed, there is some debate in the security literature about the best way to represent the actions of an adversary: are they making decisions, or do we need to assume their choices are random because of a lack of information about their objectives or knowledge or because random, opportunistic events dominate the adversary's choice of actions? We believe that both approaches have situations where they should be applied. However, in cases where it is reasonable to use a probability distribution over actions, it is critical to capture any relevant correlations in the model. For example, Figure 25 shows a scenario where the election of presidential candidate X is expected to have an impact on whether or not a terrorist attacks. On the margin, P(X is elected) = 0.6 and P(Attack) = 0.5. If we assume that an attack is more likely if X is elected, e.g. $r = 0.5$, then as shown in the lower left of Figure 25, P(Attack | X elected) = 0.704 > P(Attack). If we assume a negative association (lower right), $r = -0.35$, then P(Attack | X elected) = 0.357 < P(Attack). Assuming that these two uncertainties are independent could lead to serious errors in anticipating an attack.

It is difficult to derive closed form solutions for more than two correlated binomial uncertainties. However, it is possible to apply the NORTA approach to simulate an n dimensional standard normal vector $Z \sim N(Z_1, Z_1, \ldots Z_n)$ with correlation matrix Σ. In general, we define a binary or binomial variable $X_i = 1$ with probability p_i; $X_i = 0$ with probability $1 - p_i$, and then set $X_i = 1$ if $Z_i \leq z(p_i)$, $X_i = 0$ otherwise, where $z(p_i)$ is the p_i^{th} percentile of the standard normal distribution. Simulated data can be used to empirically calculate all of the desired conditional probabilities, e.g. $\Pr(X_i = 1 \mid X_j = 1, X_k = 0)$. For details on the process required to determine the target correlation matrix for Z,

see Emrich and Piedmonte (1991).

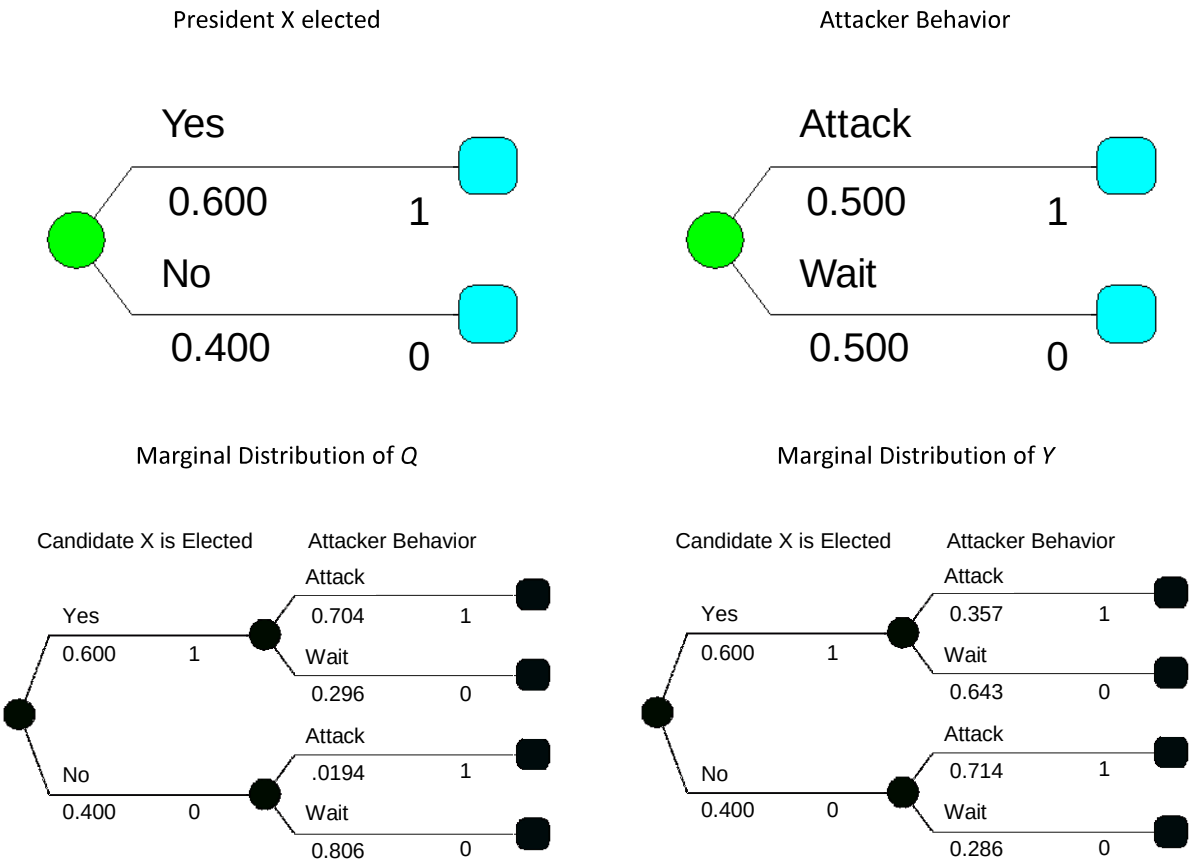
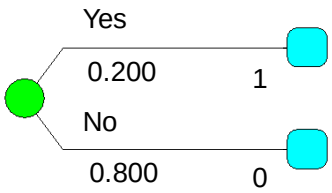
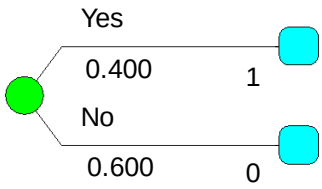


Figure 25: Example of a correlated "decision" uncertainty

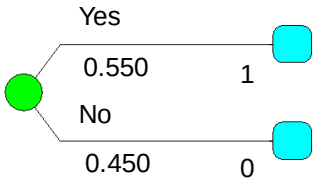
As an example, we consider three generic binomial variables, *A*, *B* and *C* as depicted in the upper panels of Figure 26. The bottom panel of Figure 26 shows the uncorrelated uncertainties and how those uncertainties are transformed given the correlation matrix in the middle panel of Figure 26 based on the average of 100,000 simulated paths through the correlated event tree rounded to two decimal places. The positive correlations have the desired effects, e.g. P(*B*=1 | *A* = 1) = 0.60 > P(*B*=1) = 0.4 and P(*C*=1 | *A*=1, *B*=1) = 0.91 > P(*C*=1) = 0.55. Many probabilistic risk analyses are conducted to find paths of high probabilities, particularly those high probability paths associated with extreme outcomes. The joint probabilities of the extreme events of P(A=1,B=1,C=1) = 0.109 when we include correlations and 0.044 when the correlations are ignored; P(A=0,B=0,C=0) = 0.339 including the correlations and 0.216 when we ignore correlation. These extreme paths are about 2.5 and 1.5 times as likely to occur, respectively, when we incorporate the correlations in Figure 26 illustrating that ignoring correlation in random events can lead to significant impacts in the likelihood estimates of paths in the tree.



Marginal Distribution of A



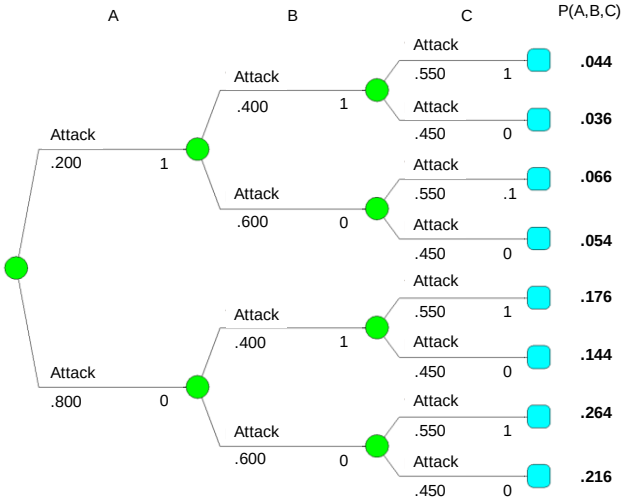
Marginal Distribution of B



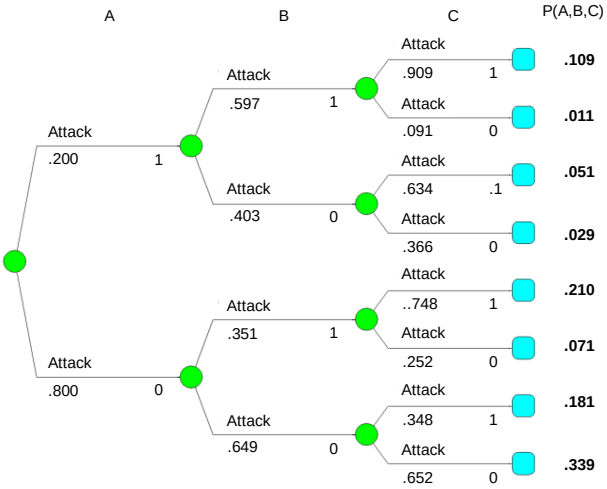
Marginal Distribution of C

A	B	C
1.00	0.20	0.25
0.20	1.00	0.40
0.25	0.40	1.00

Correlation Matrix for Uncertainties



Uncorrelated Decision Tree for 3 Uncertainties



Correlated Decision Tree for 3 Uncertainties

Figure 26: Empirical Estimate of Correlations for Three Uncertainties Based on 100,000 Simulated Triplets (joint probability of each path in bold)

The mathematical properties of correlations do place bounds on the values that are feasible and can be accommodated with the approach. Regardless of the number of uncertainties, for any pair of uncertainties *i* and *j*,

$$\max\left(-\left(p_i p_j (1-p_j) / (1-p_i)\right)^{1/2}, -\left((1-p_i)(1-p_j) p_j / p_i\right)^{1/2}\right) \le \rho_{ij} \text{ and}$$

$$\rho_{ij} \le \min\left(\left(p_i (1-p_j)(1-p_i) / p_k\right)^{1/2}, \left(p_j (1-p_i)(1-p_j) / p_i\right)^{1/2}\right)$$

5 MATHEMATICAL PROGRAMMING MODEL

The decision tree formulation has some nice features such as an intuitive structure and solution, and the ability to view suboptimal solutions if desired, and it clearly shows the sequential nature of the problem we investigate. However, there are several drawbacks that we highlight in this section. We then present an alternative solution, a mixed integer program (MIP), where binary variables are used to represent the safeguard and attack strategies of the defender and attacker, respectively.

5.1 DECISION TREE DRAWBACKS

Decision trees expand rapidly in terms of decision nodes and uncertainties, and the number of endpoints to be calculated grows quickly. For example, in the simple motivating example provided in Section 1 of this report, with two defender decisions, one attacker decision, and three uncertainties, 44 endpoints are calculated in the DPL software. That number grows to 1,296 in the larger model presented in Section 2, and that does not include the full set of defender-attacker strategies listed in Section 2.1.

In addition to becoming a computational burden, large decision trees are also difficult from the perspective of obtaining model parameters and graphically communicating the optimal policy to decision makers. It can be very time consuming to assess the conditional probabilities necessary to fill the decision tree. However, with the correlated decision tree methods used here, we only need to assess the marginal distributions and the correlations. This provides an improved method of assessing key parameters but does not resolve the other issues mentioned.

Commercial decision tree packages are limited in the size of the tree they can solve and generally do not exploit parallel computer architectures. On the other hand, a MIP formulation could potentially be solved on parallel computer architectures.

5.2 MATHEMATICAL PROGRAMMING CONSIDERATIONS

Integer programming problems are usually solved using cutting planes and variations of branch-and-bound methods. For n binary variables, the solution method could need to consider 2^n combinations. That number increases when the binary variables are changed from {0, 1} binary variables to integer-valued variables {0, 1, 2, …}. The highly constrained nature of our problem limits the dimensionality for the optimization problem since the constraints will eliminate many of the combinations that need to be considered.

For simplicity, if each safeguard is modeled as two binary decision variables X_{yes} and X_{no}, only one of these variables can be equal to one, e.g. $X_{yes} + X_{no} = 1$. The safeguard is either implemented, in which case $X_{yes} = 1, X_{no} = 0$; or it is not implemented, in which case $X_{yes} = 0, X_{no} = 1$. It is not possible that both variables equal one or zero at the same time. These ideas are further discussed in a later section.

As we mentioned, another advantage to a mathematical programming formulation is that it is relatively easier to take advantage of parallel computing resources compared to a decision tree representation. For these reasons, we use the optimization framework described in this section that can be solved in standard optimization software using the CPLEX solver. Code for the optimization software, GAMS, can be found in the appendix.

5.2.1 Objective Function Formulation

At the time of an attacker's decision, the defender has already made his choice(s) of safeguards to implement. Given the observed set of defender choices, the attacker can calculate the expected value of each of his alternatives (which target to attack, if any). If we consider the consequences as positive values (to the defender) then the attacker seeks to maximize the expected value of the outcomes whereas the defender wishes to minimize. We discuss this in more detail when we present the problem, but take a minute to discuss a related strand of literature, sequential optimization problems.

The sequential nature of a decision tree makes it easy to implement the two objective functions at the appropriate nodes: the defender seeks to minimize the expected payoffs going forward, and the attacker seeks to maximize at subsequent nodes.

Implementing two conflicting objective functions is not as straightforward in a MIP formulation. The problem appears on its face to suffer from the need to sequentially solve separate optimization problems. For instance, the defender makes a choice about the portfolio of safeguards to implement, and then the attacker makes his decision about an attack strategy given the defender's decision. However, the defender must make his initial choice given his beliefs about the attacker's subsequent strategy. In order to obtain a solution for defender and attacker, the defender's optimization problem must be solved with the attacker's optimization problem as constraints. This formulation is known as an integer bilevel linear program (IBLP).

Moore and Bard (1990) discuss the challenges in solving the class of IBLP programs, which require specialized branch-and-bound methods and can be computationally challenging to solve. The concerns in Moore and Bard (1990) and other related work from Bard (1998) and DeNegre and Ralphs (2008) arise from the fact that there are two decision makers with non-aligned objectives and two sets of decision variables. The formulation used in our approach treats each joint strategy as one variable rather than two separate dependent variables, which helps avoid some of these concerns.

5.3 Tree Formulation

The MIP formulation of the decision tree follows the work of Paruchuri, et al. (2008). To illustrate the logic, consider a small example with two decision makers, each with one choice as shown in Figure 27. Again we consider a sequential problem where the defender makes a choice and the attacker, given the defender's choice, makes his decision thereafter. In this example, the payoffs are the same for each player and are represented by the values in brackets. If the defender and attacker both choose 'yes' then the payoff is 20 to each of them. If the defender chooses 'yes' and the attacker chooses 'no' then the payoff is 10 to each player. The defender's objective is to minimize his payoff and the attacker aims to maximize his payoff.

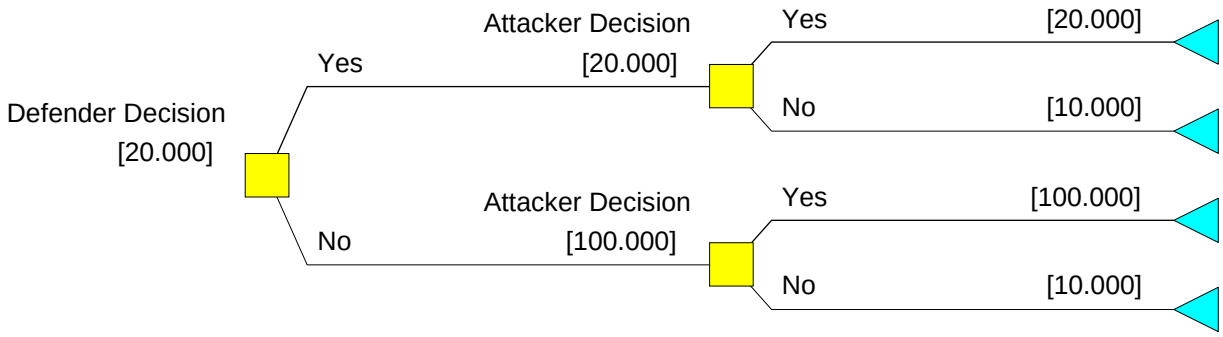


Figure 27: Defender Strategy

If the minimizing defender chooses yes, then the maximizing attacker will choose yes because 20 > 10. If the defender chooses no, then the attacker will choose yes because 100 > 10. The minimizing defender, knowing the attacker's best response to his strategy, will choose yes since 20 < 100. In the next subsection, we discuss the structure of an optimization problem that is equivalent to this decision tree.

5.4 Sets and Parameters

Following the general structure of Paruchuri et al. (2008), we have the following parameters and variables in the optimization problem.

Table 8: MIP Parameter Descriptions

Set	Description
$\boldsymbol{D}$	Set of defender strategies, indexed by d
$\boldsymbol{A}$	Set of attacker strategies, indexed by a
$\boldsymbol{\delta_d}$	= 1 defender chooses strategy d; = 0 otherwise
$\boldsymbol{\alpha_a}$	= 1 if attacker chooses strategy a; = 0 otherwise
$\boldsymbol{y_{da}}$	= 1 if defender chooses strategy d AND attacker chooses strategy a; = 0 otherwise
$\boldsymbol{\Delta_{da}}$	Matrix of defender payoffs
$\boldsymbol{\Omega_{da}}$	Matrix of attacker payoffs
$\boldsymbol{\beta}$	Decision variable used to bound payoffs in the constraints
$\boldsymbol{M}$	"Big-M" parameter

Applying the notation in Table 8 to the simple game tree in Figure 27, $D = \ yes, no$ and $A = \ yes, no$ are the defender's and attacker's action sets, respectively. So we have $d = 1, 2$ and $a = 1,2$ for the set of indices of strategies for each player.

If the defender chooses yes $(\delta_1 = 1)$, then the attacker's best response is to choose yes $(\alpha_1 = 1)$ since 20 > 10. If the defender chooses no $(\delta_2 = 1)$, then the attacker's best response is yes $(\alpha_1 = 1)$ since 100 > 10. The payoffs in our decision tree are the same as in the optimization problem. The defender's objective is to minimize the payoff and the attacker's objective is to maximize the payoff. The payoff matrix is as follows, where each cell (d, a) contains the payoffs $(\Delta_{da}, \Omega_{da})$.

	A1=yes	A2=no
D1=yes	(20,20)	(10,10)
D2=no	(100,100)	(10,10)

5.5 Mathematical Formulation

The defender solves the following problem, which is a mixed-integer quadratic program (MIQP). Paruchuri, et al. (2008) showed that their formulation could be reconfigured as a MILP, which we adapt to a MIP as shown later.

$$\min_{\delta,\alpha,\beta} \sum_{d\in D}\sum_{a\in A} \Delta_{da}\delta_d\alpha_a \qquad (1)$$

$$s.t. \quad \sum_{d\in D} \delta_d = 1$$

$$\sum_{a\in A} \alpha_a = 1$$

$$0 \le \beta - \sum_{d\in D} \Omega_{da}\delta_d \le 1 - \alpha_a \; M \qquad \forall a \in A$$

$$\delta_d \in 0,1$$

$$\alpha_a \in 0,1$$

$$\beta \in R$$

The expression $\delta_d\alpha_a$ in the objective function is equal to one when the defender chooses strategy $d \in D$ and the attacker chooses strategy $a \in A$. This introduces a nonlinearity that results in a mixed-integer quadratic program (MIQP). We can introduce a change of variable to reformulate the problem as a MIP by considering a binary variable, y_{da}, which takes on the value 1 if strategy d is chosen by the defender and strategy a is chosen by the attacker; otherwise the $y_{da} = 0$. In this new formulation, the defender's decision isn't explicitly modeled as a choice variable, but the attacker's choice, α_a, is and the objective function and several constraints are summed over the full set of the defender's strategies, D. The joint decision of choosing α_a and y_{da} determines the defender's choice.

$$\min_{\alpha,y,\beta} \sum_{d\in D}\sum_{a\in A} \Delta_{da}y_{da} \qquad (2)$$

dummy's × payoffs to defender

s.t. $\sum_{d\in D}\sum_{a\in A} y_{da} = 1$

only one dummy = 1 -> only one path

$\alpha_a \le \sum_{d\in D} y_{da} \le 1 \qquad \forall a \in A$

if $\alpha_a = 1$ then then defender path must include a; if $\alpha_a = 0$ then path could or could not include a

$\sum_{a\in A} \alpha_a = 1$

attacker must choose exactly one action

$0 \le (\beta - \sum_{d\in D} \Omega_{da}(\sum_{\gamma\in A} y_{\delta\gamma})) \qquad \forall a \in A$

$\beta \ge$ Attacker Payoff given for actions a and d[1]

[1] Note that γ is used in place of a here as a change of variable since there is an equation for each $a \in A$.

$$\beta - \sum_{d \in D} \Omega_{da} \left(\sum_{\gamma \in A} y_{\delta\gamma} \right) \leq 1 - \alpha_a \; M \quad \forall a \in A$$

If attacker plays α_a then β is the attacker payoff for action a

$y_{da} \in \{0,1\}$
$\alpha_a \in \{0,1\}$
$\beta \in R$

Domain constraints

5.5.1 Objective Function

If $y_{da} = 1$, then the joint strategy (d, a) is chosen, the defender chooses strategy $d \in D$, and the attacker chooses strategy $a \in A$. In that case, the defender will realize the payoff Δ_{da} from his payoff matrix Δ. If another strategy is chosen then Δ_{da} does not contribute to the objective function. To relate this to the decision tree formulation, the choice of y_{da} is equivalent to a path on the tree that represents a unique selection of defender and attacker strategies. Each path is represented by one y_{da} variable.

5.5.2 Constraints

The set of constraints serve to guarantee that the attacker's choice, given the defender's decision, is optimal for the attacker. Knowing this sequence of events, the defender's decision is maximized by considering what the optimal attacker choice is following the defender's action.

5.5.2.1 One joint strategy

Since the attacker and defender are each choosing a single strategy, it naturally follows that only one joint strategy can result. Therefore, only one of the y_{da} variables can be equal to one; the rest must be equal to zero. Relating this to the tree again, it is equivalent to enforcing that one and only one path through the tree is chosen.

$$\sum_{d \in D} \sum_{a \in A} y_{da} = 1$$

In our small example, this constraint can be explicitly written as:

$$y_{11} + y_{12} + y_{21} + y_{22} = 1$$

5.5.2.2 Attacker and Joint Strategies

If the attacker chooses a particular strategy a', then it must be the case that for some defender choice d there is one $y_{da'} = 1$. This constraint binds the attacker and defender choices in the y_{da} variable. This constraint applies to each attack option 'a'.

$$\alpha_a \leq \sum_{d \in D} y_{da} \leq 1$$

To illustrate using the simple example again, the constraints would become:

$$\alpha_1 \leq y_{11} + y_{21} \leq 1$$
$$\alpha_2 \leq y_{12} + y_{22} \leq 1$$

If the attacker chooses strategy 1, then the constraints appear as follows:

$$1 \leq y_{11} + y_{21} \leq 1$$
$$0 \leq y_{12} + y_{22} \leq 1$$

Clearly, the first constraint implies that y_{11} or y_{21} must be 1 and the possible set of optional strategy pairs has been reduced to two. That must be the case given how the y_{da} variable has been defined.

5.5.2.3 ATTACKER OPTIMALITY

The defender makes his choice knowing what the attacker will do following his decision. Since the attacker has a different objective from the defender, this problem must be embedded in the constraints of the program. The attacker will choose the outcome that is optimal, given the defender's decision. Using a choice variable, β, to represent the attacker's payoff, we have the following set of constraints for each potential attacker decision $a \in A$. The right-hand side of the constraint uses the big-M method to selectively enforce an upper bound in the inequality.

$$0 \leq \beta - \sum_{d \in D} \Omega_{da} \left(\sum_{\gamma \in A} y_{d\gamma} \right) \leq 1 - \alpha_a \; M \quad \forall a \in A$$

This constraint is easy to explain using our simple example. With the set of attacker decisions being {yes=1, no= 2} there are two constraints to consider:

$$0 \leq \beta - [\Omega_{11} \; y_{11} + y_{12} \; + \Omega_{21}(y_{21} + y_{22})] \leq (1 - \alpha_1)M$$
$$0 \leq \beta - \; \Omega_{12} \; y_{11} + y_{12} \; + \Omega_{22} \; y_{21} + y_{22} \quad \leq \; 1 - \alpha_2 \; M$$

Let us consider the case where $\alpha_2 = 1$, which implies $\alpha_1 = 0$. Since M is sufficiently large, it imposes no forceful upper bound on the right-hand side of the first constraint where the RHS $= \; 1 - 0 \; M = M$. It is essentially an inactive constraint as the upper bound is too large to restrict any action. In the second constraint, the RHS of the inequality is restricted to $\; 1 - 1 \; M = 0$ which, together with the LHS implies equality. Therefore, we have

$$\beta = \; \Omega_{12} \; y_{11} + y_{12} \; + \Omega_{22} \; y_{21} + y_{22}$$

Since the attacker strategy is no $(\alpha_2 = 1)$, if the defender chooses yes, then $\beta = \Omega_{12} \; y_{11} + y_{12}$, and if the defender chooses no, then $\beta = \Omega_{22} \; y_{21} + y_{22}$. We know that $y_{12} = y_{22} = 0$ since the only joint strategy that can be 1 is of the form y_{d1}. Since they are binary variables, Ω_{da} represents the attacker's payoff if the combined strategy is $\; d, a \;$, so we see that β is actually the attacker's payoff for the choice of α_a. Similarly, if the attacker chose yes, then $\beta = \Omega_{11}$ or $\beta = \Omega_{21}$ depending on the defender's choice.

5.5.2.4 DOMAIN CONSTRAINTS

The last set of constraints enforces the restriction that each y_{da} and α_a is binary and declares that β is a free variable. It is straightforward to see how these constraints can be used in the small example.

$$y_{da} \in \; 0,1 \quad \forall \; d, a \; \in D \text{ x } A$$
$$\alpha_a \in \; 0,1 \quad \forall a \in A$$
$$\beta \in R$$

5.6 RESULTS

5.6.1 DECISION TREE FORMULATION

The optimal policy in the decision tree is the defender chooses yes and the attacker chooses yes for a payoff of 20 to the defender. As we discussed before, the defender can choose yes (strategy 1) with an expected payoff of 20 based on the attacker's optimal response or the defender can choose no (strategy 2) and earn a payoff of 100. The tree in Figure 28 clearly shows the optimal choice as the highlighted path.

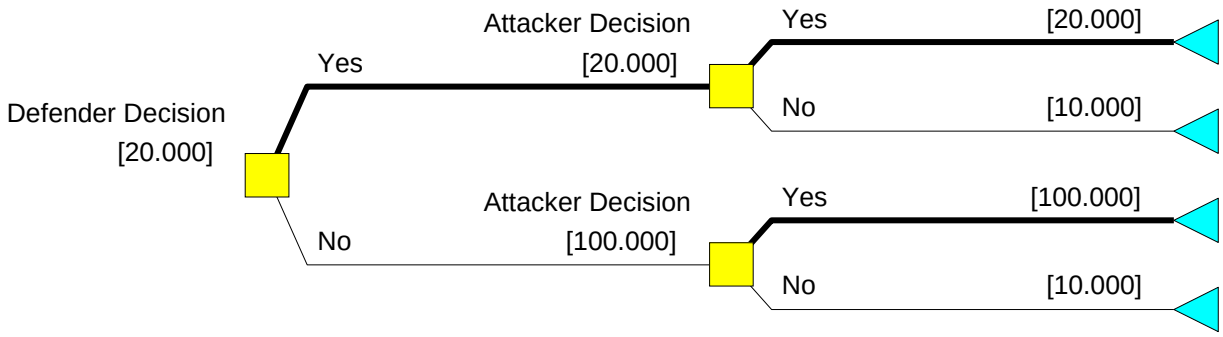


Figure 28: Policy Tree for MIP Example

5.6.2 Optimization

Table 9 shows that the variable $y_{11} = 1$ which indicates that the optimal joint strategy is for the defender to choose yes and the attacker to choose yes. This is supported by the fact that $\alpha_1 = 1$. Notice that $\beta = 20$; this is the payoff earned by the attacker under this scenario.

Table 9: MIP results for small example

y_{11}	y_{12}	y_{21}	y_{22}	α_1	α_2	β
1	0	0	0	1	0	20

5.6.3 Motivating Example

We can follow the same logic and structure just presented to solve the motivating example provided in Section 1 of this report. For convenience, we show Figure 5 again below as Figure 29. What we will look for in the optimization results are decision variables that represent the defender choice (Yes, No), the attacker choice (Yes), and the defender's expected loss of 23.335.

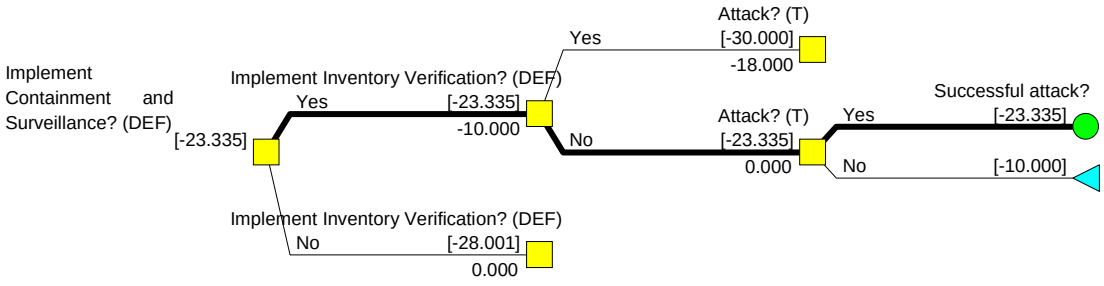


Figure 29: Motivating Example - Policy Tree

There are four defender strategies, $D = YY, YN, NY, NN$, and again just two attacker strategies, $A = \{Y, N\}$ where Y and N are used to abbreviate the choices yes and no. Figure 29 does not show the D = NY and NN choices, for compactness.

We solved the simple problem formulation earlier using the Microsoft Excel solver because of the relatively small nature of the problem. In this example, we have far more endpoints and so solving in Excel would be tedious. Using the algebraic modeling language GAMS (www.gams.com) makes it easier to model these larger problems. We still use Microsoft Excel, however, to provide initialization values in DPL and also to link this information to GAMS. Entering parameters in a spreadsheet is easier than modifying the parameters directly in DPL, and the same applies to modifying the GAMS code.

The formulation is similar to the simple example in terms of variable definitions. The mathematical formulation is the same, but there will be more equations when not written in the compact algebraic form.

The full set of constraints for the motivating example is shown below and helps to illustrate the use of an algebraic modeling system and a specialized large-scale MIP solver.

$$\min_{y,\alpha,\beta} \Delta_{11}y_{11} + \Delta_{12}y_{12} + \Delta_{21}y_{21} + \cdots + \Delta_{41}y_{41} + \Delta_{42}y_{42} \qquad (3)$$

$$s.t.\ y_{11} + y_{12} + \cdots + y_{42} = 1$$

$$\alpha_1 \le y_{11} + \cdots + y_{41} \le 1$$

$$\alpha_2 \le y_{12} + \cdots + y_{42} \le 1$$

$$\alpha_1 + \alpha_2 = 1$$

$$0 \le (\beta - \Omega_{11}\ y_{11} + y_{12} + \Omega_{21}\ y_{21} + y_{22} + \cdots + \Omega_{41}\ y_{41} + y_{42} \le 1 - \alpha_1\ M$$

$$0 \le (\beta - \Omega_{12}\ y_{11} + y_{12} + \Omega_{22}\ y_{21} + y_{22} + \cdots + \Omega_{42}\ y_{41} + y_{42} \le 1 - \alpha_2\ M$$

$$y_{11}, y_{12}, y_{21}, \ldots, y_{42} \in 0,1$$

$$\alpha_1, \alpha_2 \in 0,1$$

$$\beta \in R$$

The problem solved in 0.014 seconds according to the GAMS output, using a Lenovo ThinkPad X230 laptop with an Intel Core i7-3520M processor @2.9 GHz. The results are provided in Table 10.

Table 10: Optimization results small model

Variable	Objective	α_1	α_2	β				
Value	23.335	1	0	13.335				
Variable	y_{11}	y_{12}	y_{21}	y_{22}	y_{31}	y_{32}	y_{41}	y_{42}
Value	0	0	1	0	0	0	0	0

The results here match the results from the decision tree. The optimal strategy is for the defender to choose strategy 2 {YN} and for the attacker to choose strategy 1 {Y}. This results in an expected payoff to the defender of 16.669 as we found in the decision tree formulation using DPL (Figure 29). The attacker's payoff is captured in the decision variable β. In the previous example, we had a symmetric payoff structure. The difference here is that the defender incurs a cost to use a yes strategy. The first safeguard costs 10 units and the second safeguard costs 18 units. These costs are shown in the decision tree in Figure 29 below the branch extending from each decision node.

For example, if the defender chooses yes in the first decision, you see a cost of 10 shown underneath the 'yes' branch and a cost of 0 below the 'no' branch.

5.7 LARGE SCALE RUNS

To assess the computational burden of the mathematical programming formulation, the problem was scaled to consider a larger number of defender strategies and attack scenarios than previously used in the decision tree formulations. Notional values were used to simulate payoffs for various attack scenarios.

5.7.1 PRE-PROCESSING

In this section, we give an overview of the pre-processing performed for the runs in order to show how we ran the larger problems.

5.7.1.1 STRATEGIES

A vector of safeguard decisions (1=yes, 0=no) was generated to represent the sequence of decisions by the defender regarding which safeguards to implement. For example, with three safeguards, the strategies are represented as follows in Table 11.

Table 11: Strategy Vectors

Strategy	Safeguard 1	Safeguard 2	Safeguard 3	Vector
1	No	No	No	000
2	No	No	Yes	001
3	No	Yes	No	010
4	No	Yes	Yes	011
5	Yes	No	No	100
6	Yes	No	Yes	101
7	Yes	Yes	No	110
8	Yes	Yes	Yes	111

Clearly, these vectors get quite large when there are more safeguards considered. Generally speaking, each vector for strategy d could be considered as $X^d = [X_1^d\ X_2^d \dots X_s^d]$ where there is one element for each safeguard. These variables are used in a pre-processing program that generates the probability parameters and strategy costs for the GAMS model.

5.7.1.2 UNCERTAINTIES

The matrix P_{sa} contains probability assessments for the probability of safeguard s detecting attack a. As usual, these are completely notional and were arbitrarily assigned to generic safeguards and attack scenarios. These values were used in combination with the strategy vectors to generate P_{da}, the probability of a successful diversion with the joint strategy (d, a). Table 11 shows the matrix for 8 potential safeguards (rows, the first 3 of which correspond to the three safeguards in Table 10) and 10 potential attack scenarios (columns).

Table 12: Sample P_{sa} matrix

0.9	0.8	0	0.9	0.8	0	0.9	0.8	0	0.9
0.8	0	0	0.9	0	0	0.9	0	0	0.9
0.7	0.6	0.7	0.8	0.6	0.7	0.8	0.6	0.7	0.8
0.9	0.8	0	0.9	0.8	0	0.9	0.8	0	0.9
0.8	0	0	0.9	0	0	0.9	0	0	0.9

0.7	0.6	0.7	0.8	0.6	0.7	0.8	0.6	0.7	0.8
0.7	0.6	0.7	0.8	0.6	0.7	0.8	0.6	0.7	0.8
0.9	0.8	0	0.9	0.8	0	0.9	0.8	0	0.9

To calculate P_{da} we use the formula from earlier, $P_{da} = \quad_{s \in S}(1 - P_{sa})$, but with a slight modification. Previously, we mentioned that $P_{sa} = 0$ when the safeguard is not deployed and is otherwise in the set [0, 1] depending on the assessment. To calculate this algorithmically, we used $P_{da} = \quad_{s \in S}(1 - P_{sa} * X_s^d)$ with $X_s^d = 1$ when safeguard s is used in strategy d and equals zero otherwise. This results in a matrix such as the one shown in

Table 13, in which the columns represent the 10 potential attack scenarios and the rows represent the three safeguards (8 strategies) considered by the defender. Note the top row is all ones as this is the strategy in which no safeguards are used at all (so all attacks are successful).

Table 13: Sample P_{da} matrix

1	1	1	1	1	1	1	1	1	1
0.3	0.4	0.3	0.2	0.4	0.3	0.2	0.4	0.3	0.2
0.2	1	1	0.1	1	1	0.1	1	1	0.1
0.06	0.4	0.3	0.02	0.4	0.3	0.02	0.4	0.3	0.02
0.1	0.2	1	0.1	0.2	1	0.1	0.2	1	0.1
0.03	0.08	0.3	0.02	0.08	0.3	0.02	0.08	0.3	0.02
0.02	0.2	1	0.01	0.2	1	0.01	0.2	1	0.01
0.006	0.08	0.3	0.002	0.08	0.3	0.002	0.08	0.3	0.002

5.7.1.3 COSTS

Using the same vector information, the strategy costs are generated in the program. Starting with a cost of 0, the strategy cost is iteratively calculated as follows, where C_d is the cost of strategy *d*, C_s is the cost of safeguard s, and X_s^d is the binary variable representing whether or not safeguard s is included in strategy d.

$$C_d = C_d + C_s X_s^d$$

The number of strategies grows rapidly with an increase in the number of safeguards included, with a total of 2^S strategies, where S is the number of safeguards.

5.7.2 RESULTS

The number of equations used in the GAMS commercial optimization software represents both constraints and calculated equations. Constraints are the same as appear in the model formulation and take the form $\boldsymbol{Ax} = \boldsymbol{b}, \boldsymbol{Ax} \leq \boldsymbol{b}, \text{and } \boldsymbol{Ax} \geq \boldsymbol{b}$**.** The calculated equations are expressions such as the objective function. The number of equations is linear in the number of attack strategies, a, considered for the model. There are four constraints that are evaluated "for all a" in addition to the objective function and the constraints restricting the attacker to one choice and only one joint strategy. Therefore, the number of equations grows as $O(a)$ using big-O notation.

The number of variables is a function of both the number of safeguards (and therefore strategies) and the number of attack scenarios. The number of y_{da} joint strategy variables is the dominating factor in the growth of the number of variables, while the rest of the growth is linear in a, resulting in a growth rate of $O\ 2^s$.

Non-zero elements are members of the matrices of payoffs, coefficients, costs and other parameters used in the model. As the number of strategies and attack scenarios grow, so will the number of non-zero elements.

Run-time is the amount of time it takes GAMS to initiate the run by reading in data files, performing pre-processing steps, and the time it takes CPLEX to solve the problem and report the answer. It does not include pre-processing time to create the parameters for the run. A separate program was written to provide input data for the runs. This information includes P_{da}, the probability of a successful attack when joint strategy *(d, a)* is chosen by the defender and attacker. It also includes the cost of each strategy, which is determined by cost assessments for each safeguard and the number of safeguards included in the strategies. This program uses other assessments to calculate these parameters.

The runs were performed using a Lenovo ThinkPad X230 with a Core i7-3520 M processor, 2.9 GHz Intel Processor, 180 GB SSD, 4MB RAM, and a 64-bit Windows 7 operating system. GAMS build 23.9.5 was used and has a full license including CPLEX. The results are shown in Figure 30. Attack options refers to the number of thefts or diversions considered by the attacker. The number of safeguards considered appears on the horizontal axis. The number of defender strategies could be obtained from this by calculating 2^s.

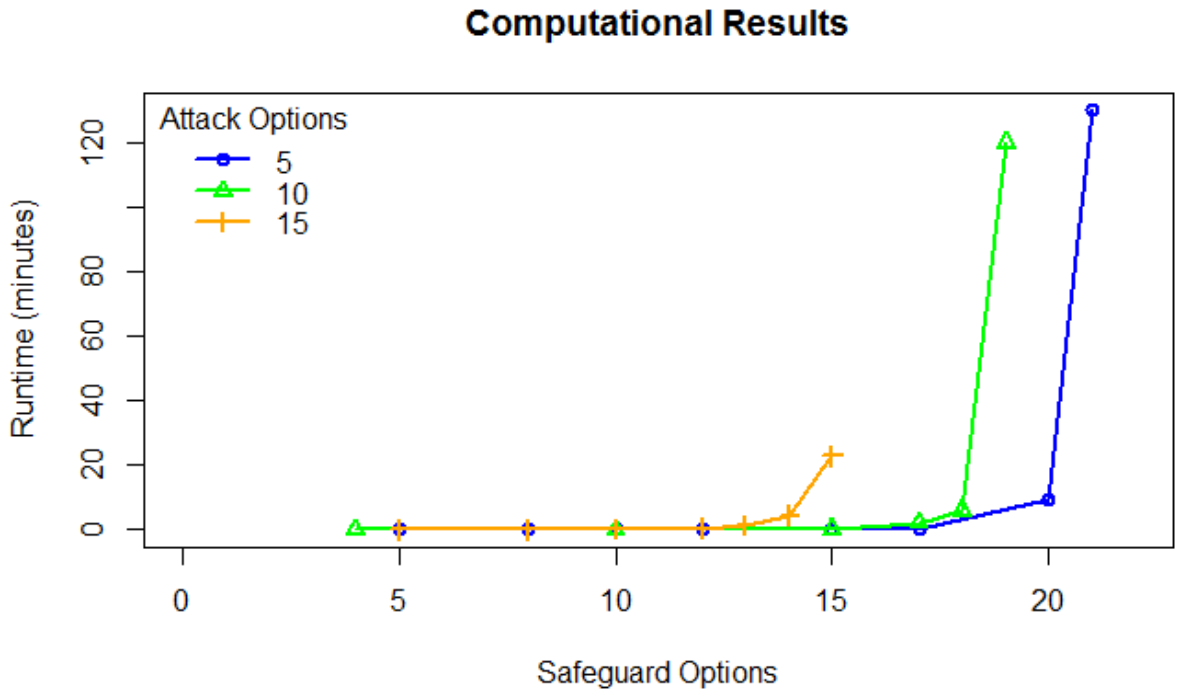


Figure 30: Computational results of MIP for various defender and attacker strategies

For an equal number of attack scenarios with the same payoffs and the same probability assessments, the objective function value improves (non-increasing for the defender) as more safeguards are considered. This makes sense because, as new safeguards are considered, all of the previous options are still available. For example,

if there are 4 safeguards considered in run 2 and we add 6 more safeguards in run 3, we still could choose the optimal strategy (using just those four safeguards) and choose not to use any of the new 6 safeguards. This should result in the same objective function. Therefore, it should be non-increasing in the number of safeguards, provided the same set of attack scenarios (and associated parameter values) is used.

This computational exercise was to test the algorithm using a personal computer. Parallel processing and supercomputing would enable the solution of larger problems in reasonable times. In addition, real world applications are likely to have more than 15 possible attacks and 20 possible safeguard options. Modeling techniques such as prioritization of strategies to include only the most effective, or aggregating them into classes of strategies, could be used to address the computational challenge of large-scale analysis.

Finally, the computational burden could be reduced by the inclusion of a budget constraint. A sufficiently tight budget constraint could restrict the use of many safeguard combinations (i.e. strategies). This could be accomplished as a single constraint in the mathematical programming formulation where I is the budget level.

$$\sum_{d \in D} y_{da} c_d \leq I$$

It could also be done in pre-processing when the parameters are generated, by eliminating a strategy from consideration once the strategy cost exceeds the budget threshold. This might actually be computationally more efficient. At that point, adding more safeguards is not really useful. It may be practical to consider only strategies with a relatively small number of safeguards. This could dramatically reduce the computational burden.

6 FURTHER WORK

In this section we present some opportunities for further exploration relating to methodology, extensions, and computational feasibility.

6.1 MULTIPLE FACILITIES AND PERIODS

The results presented in this report demonstrate our methodology on one facility, but this could be extended to allocate resources across multiple facilities through a budget constraint tied across the facilities. To allocate resources across multiple facilities, the constraint would take the form below, where F is the set of facilities.

$$\sum_{f \in F} \sum_{d \in D} \sum_{a \in A} c_d^f y_{da}^f \leq I$$

Exploring multi-stage problems where the budget could be allocated over multiple periods would also be an interesting extension. An example could be a new safeguard technology that takes one period to develop to operational capability, requiring an investment in the previous period to deploy. In period 1, the defender could make an R&D investment, with the safeguard implemented in the second period. As a second option, the defender could spend the resources on a different immediately available safeguard in the first period. A third option could be for the defender to save the funds for the second period and wait to see if the attacker has exploited a vulnerability in the first period, and then target the investment in the second period to mitigate the exposed risk. The integral of risk over time could also be used as a metric for guiding R&D investments.

6.2 STRATEGIES

We assumed in this model that the list of attacker strategies would be exhaustive, based on the facility's design basis threat (DBT). However, it would be worth studying, particularly in the context of a multi-period game, how to incorporate an unanticipated attack strategy. This might be especially relevant in the context of the insider threat, because the malicious insider may be able to identify vulnerabilities not generally known to the facility's security personnel. A potential approach could tie into the previously mentioned idea of considering perceived probabilities. The attacker may have a lower probability of detection assessment in some areas due to a perceived vulnerability.

6.3 ALTERNATIVE SOLUTION APPROACHES

Depending on the actual probability assessments for the safeguards, it could be the case that the probability of a successful diversion becomes very small with the introduction of just a few safeguards. Consider, as an example, four safeguards with a probability of detection of 0.8 each for some particular attack strategy. The probability of a successful diversion would be $1 - 0.8^{\ 4} = 0.0016$. If this were the case, then it might not be necessary to consider more safeguards if the attacker wouldn't attack with such a low probability of success. Therefore, it may be worth considering only strategies for which the probability of successful diversion is above some threshold.

Considering a directed graph or network representation could also be worth exploring to see if it yields faster solution times. The defender would deploy safeguards at edges on the graph that impose a cost to the attacker, and the attacker would seek the lowest-cost path through the network.

6.4 Computational Efforts

One of the most important and immediate extensions of this work should be to parallelize the algorithms to be run on supercomputers and other advanced computing resources available to the DOE national laboratories. Not only will this allow for larger models to be run, but it can also identify the largest models that can be reasonably solved in a supercomputing environment.

Additional research on reducing the computational burden due to the large number of strategies is worthwhile. The mixed integer programming formulation may have a specialized structure that could allow for faster solution times than the branch and cut methods used by the CPLEX solver. Focused research on integer programming techniques may yield opportunities to reduce solution times for larger problems.

Despite the advanced computing resources available to the DOE national laboratories, it would be worth examining solutions to problems that are arbitrarily larger than supercomputers could solve to optimality. A sampling strategy that repeatedly solves "relaxed" MIPs using randomly sampled safeguards from the full safeguard set could potentially yield reasonable approximations to the unrelaxed problem. There could be insights gleaned on the safeguards that frequently appear in the optimal solutions. Furthermore, learning algorithms could identify safeguards that tend to appear together. For example, we could find that three particular safeguards appear in more than 90% of the optimal relaxed solutions. As a further example, we could find that item counting appears in nearly all cases in which containment and surveillance appears, and we could also see that item counting rarely appears in solutions along with inventory verification. Constraint sampling has been proposed to solve some difficult dynamic programming problems, and there may be some analogous methods to be applied here.

6.5 Parameter Assessments

Another interesting aspect that could be explored further is the probability of a successful diversion. We assumed knowledge of these probabilities for both the defender and attacker, which is reasonable given that a malicious insider adversary would be likely to have some understanding of the safeguards. However, it would be interesting to capture "perceived probabilities." The attacker may have some notion of the likelihood of defeating a safeguard, but that estimate might be different from the estimates the defender has. In reference to the previous recommendation of examining multi-period games, this could also provide an opportunity for both players to refine their probability assessments given the attack option executed in the first period (including whether or not it was successful).

The uncertainties regarding a successful diversion were expressed in terms of the quality of material stolen and the yield from such a theft. However, these could be easily modified to express further uncertainties. For example, there may be uncertainty over the attacker's capabilities to generate a weapon, the particular adversary group that is responsible, potential recovery of the material, and other uncertainties deemed important by DOE. This is a case where the correlated uncertainties introduced in this report would be advantageous. Rather than eliciting many conditional probabilities, these additional uncertainties need to be expressed only in terms of their marginal distributions and pairwise correlations.

In order to make a model such as the one presented in this report operational, considerable time will need to be given by subject matter experts to assess the probabilities and other parameters needed for the model. Specifically, we would need to elicit probability distributions of the quality of the material stolen for each target to be considered as well as the associated yield from any weapon developed using that material. With the use of copulas, we can dramatically reduce the number of assessments needed. Instead of assessing the quality

distribution and then a conditional assessment based on each quality outcome, we can instead elicit the marginal distributions and pairwise correlations. Nonetheless, it might be worth doing a formal comparison of these two alternative elicitation techniques to document the advantages and disadvantages of each in the actual SME elicitation environment.

Finally, we have used a nominal UREX+ facility here with safeguards and attack scenarios developed using open-source materials. Subject matter experts including security professionals, scientists, and others would need to determine the actual safeguards available, identify possible targets, and assess the potential vulnerabilities, threats, and consequences associated with those targets. Mixed Oxide (MOX) fuel fabrication facilities (MFFF) have been under construction to deal with large quantities of weapons-usable highly enriched uranium and plutonium following the end of the Cold War (NRC 2005). These excess materials will be converted into fuel for nuclear reactors. New facilities and configurations could result in enhanced security but also new vulnerabilities. A next step could be to analyze an actual facility such as a MFFF and elicit the previously mentioned parameters. This could require access to non-open source information, however.

7 CONCLUSIONS

In this model setup, we were able to solve a Stackelberg game using a decision tree and an equivalent MIP formulation. The advantage of the MIP formulation is that while a decision tree is a logical, intuitive way to represent a decision problem, the reformulation as a mixed integer program provides a way to solve a much larger problem with substantially more strategies for both the defender and attacker. The MIP formulation also provides a way to parallelize the solution if the number of variables becomes too large.

We also considered that the uncertainties resulting from the quality of material stolen and the yield payoff from such diversions could be correlated. This enables subject matter experts to make initial assessments on the potential yields and qualities of various special nuclear materials (SNM) and then assess the correlations. This is much less burdensome than the number of conditional probability assessments that would have to be made otherwise. As technologies and capabilities improve, the decision maker can reassess the initial distributions without the cumbersome task of updating all of the conditional probabilities.

Given probability and cost assessments from subject matter experts, policy makers can use this model to inform the decision-making process on which safeguards can be implemented to best mitigate the terrorist threat to the commercial nuclear fuel cycle. There are a number of safeguard options that can be deployed, either as new technologies or systems, or as add-ons to additional safeguard measures. With appropriate assessments, this model formulation provides guidance on the optimal strategies for various sets of safeguards and shows how the risk and payoff changes as a result.

8 REFERENCES

Avramidis, A. N.; Channouf, N; and L'Ecuyer, P. 2009. Efficient Correlation Matching for Fitting Discrete Multivariate Distributions with Arbitrary Marginals and Normal-Copula Dependence. *INFORMS Journal on Computing* 21(1): 88-106.

Bard, J. F. 1998. *Practical Bilevel Optimization: Algorithms and Applications.* Boston: Kluwer Academic Publishing.

Cario, M. C. and Nelson B. L. 1997. Modeling and generating random vectors with arbitrary marginal distributions and correlation matrix. Working paper.

Clemen, R. T. and Reilly, T. 1999. Correlations and Copulas for Decision and Risk Analysis. *Management Science*. 45(2) 208–224.

Clemen, R. T. and Reilly, T. 2000. Making Hard Decisions with DecisionTools, Duxbury Press, 2nd ed.

Clemen, R. T.; Fisher, G. W.; and Winkler, R. L. 2000. Assessing Dependence: Some Experimental Results. *Management Science*. 46(8) 1100–1115.

DeNegre, S. and Ralphs, T. K. 2009. "A Branch-and-Cut Algorithm for Bilevel Integer Programming," in *Proceedings of the Eleventh INFORMS Computing Society Meeting*, pp. 65-78.

Durst, et al. 2007. *Advanced Safeguards Approaches for New Reprocessing Facilities*, PNNL Report 16674.

Emrich, L. J. and Piedmonte, M. R. 1991. A Method for Generating High-Dimensional Multivariate Binary Variates. *The American Statistician,* 45 302-304.

Ezell, B. C; Bennet, S. P.; Von Winterfeldt, D.; Sokolowski, J.; and Collins, A. J. 2010. "Probabilistic Risk Analysis and Terrorism Risk." *Risk Analysis* 30, No. 4 (April 2010): 575-589.

Hammond, R. K. and Bickel, J. E. 2012. Reexamining Discrete Approximations to Continuous Distributions. *Decision Analysis*, forthcoming.

Keefer, D. L. and Bodily, S. E. 1983. Three-point Approximations for Continuous Random Variables. *Management Science* 29(5) 595-609.

Kruskal, W. H. 1958. Ordinal Measures of Association. *Journal of the American Statistical Association*, 53, 814-861.

Maurer, S. M. 2009. *WMD Terrorism – Science and Policy Choices*, MIT Press.

Moore, J. T. and Bard, J. F. 1990. The Mixed Integer Linear Bilevel Programming Problem. *Operations Research* 38(5), 911-921.

Paruchuri, P.; Pearce, J. P.; Janusz, M.; Tambe, M.; Ordonez, F.; and Kraus, S. 2008. "Efficient Algorithms to Solve Bayesian Stackelberg Games for Security Applications." Published Articles & Papers. Paper 47. http://research.create.usc.edu/published_papers/47.

Parnell, G. S.; Smith, C. M.; and Moxley, F. I. 2010. "Intelligent Adversary Risk Analysis: A Bioterrorism Risk Management Model." *Risk Analysis* 30, No. 1: 32-48.

Pereira, C. 2008. *UREX+ Process Overview*, Topical Seminar Series on Nuclear Fuel Separation Processes.

Proliferation Resistance and Physical Protection Evaluation Methodology Expert Group of the Generation IV International Forum (PRPP). 2006. *Evaluation Methodology for Proliferation Resistance and Physical Protection of Generation IV Nuclear Energy Systems*. Revision 5.

Todd, Terry. Spent Nuclear Fuel Reprocessing. Proc. of Nuclear Regulatory Commission Seminar, Rockville, MD. Idaho National Laboratory, March 25, 2008. Print.

U.S. Nuclear Regulatory Commission. Accessed March 2012. *Backgrounder on Mixed Oxide Fuel*. http://www.nrc.gov/reading-rm/doc-collections/fact-sheets/mox-bg.html.

Wang, T. and Dyer, J. S. 2012. A Copulas-Based Approach to Modeling Dependence in Decision Trees. *Operations Research*. 60:225-242.

Ward, R. 2012. Dissertation (forthcoming): A Game Theoretic Approach to Safeguards Selection and Optimization.

Wood, R. K., 2011, "Bilevel Network Interdiction Models: Formulations and Solutions," in *Wiley Encyclopedia of Operations Research and Management Science*. DOI: 10.1002/9780470400531.eorms0932 .

9 APPENDICES

9.1 APPENDIX A: DESCRIPTIONS OF DEFENDER STRATEGIES AT A REPROCESSING PLANT (WARD, 2012)

A. Item Counting

Item counting should identify the absence of discrete items, such as spent fuel rods or solid TRU product ingots.

B. Inventory Verification

Inventory verification takes place twelve times a year: eleven monthly, interim verifications (IIV) and one annual, physical inventory verification (PIV). During the PIV, the plant is shut down and flushed out to account for as much material in process hold-up as possible. During this time, any material that has been diverted should be identified as missing; however, it is possible at large facilities for diverted material to be hidden in MUF (material unaccounted for), which is the material that falls within standard measurement errors. Thus inventory verification has the potential to measure any diversion scenario, although the actual detection probability will depend on the amount of material stolen and the error associated with measurements. Inventory verification will not detect falsified spent fuel declarations.

C. Design information verification

During design information verification, inspectors ensure that physical features of the facility are consistent with declared design specifications. The 3-Dimensional Laser Range Finder Detector is used to help inspectors find anomalous plant features or features that have changed since the last inspection. This safeguard should be able to detect the attachment of any pipes or valves used to divert material from a process vessel.

D. Non Destructive Assay, NDA (Gross neutron counting; Pu/Cm-242 ratio counting)

Standard non-destructive assay techniques are used to verify shipper declarations of spent fuel composition, as well as to monitor material and approximate TRU content. In this case, it is assumed that traditional NDA techniques are applied to tanks once the tanks have reached a certain volume. The implication of this assumption is that NDA techniques will not detect the diversion of material, but they will detect diversion with replacement, because the resultant solution that is being measured will be diluted by replacement material (e.g. HNO_3).

E. Destructive Analysis, DA

Destructive analysis is used to determine the isotopic composition of a solution. As such, DA can detect diversion of material and replacement with another material. DA may also be able to detect false declarations of spent fuel composition.

F. Containment and Surveillance, C/S (cameras and directional radiation detectors)

Containment and surveillance is installed around the facility to ensure the proper, undisturbed flow of materials. C/S is also of particular utility in storage areas, where little movement or change in scenery is expected. As such, C/S can detect diversion from feed storage or from TRU product storage. C/S also has some ability to detect any diversion from the process stream, although this ability may depend on the visibility of areas accessed by the diverter and the visibility of any requisite equipment.

G. Solution Measurement and Monitoring System, SMMS

The SMMS measures process conditions such as density, volume, and mass. This monitoring system can detect the diversion of material without replacement because of decreases in mass and volume. It may also be able to detect diversion of material with replacement due to changes in density.

H. Plutonium Inventory Measurement System, PIMS

PIMS is a network of neutron detectors designed to provide Near Real-Time Accountancy (NRTA) of plutonium and process conditions. This safeguard should be able to detect any loss of plutonium in solution due to diversion.

I. Hybrid k-edge/XRF densitometry

Like many of the safeguards above, the hybrid k-edge densitometer is designed for online TRU measurements. This technology may be able to achieve greater sensitivity than current techniques, which would reduce the MUF and make protracted diversions more difficult to perpetrate without detection. This safeguard should be able to detect any loss of TRU in solution due to diversion.

J. Lead slowing-down spectroscopy

Lead slowing-down spectroscopy is a safeguard under development for the direct measurement of plutonium in spent fuel. This technique may be able to measure spent fuel composition with a higher degree of accuracy and sensitivity than current NDA techniques. As such, this tool should be able to detect falsification of spent fuel composition declarations.

9.2 APPENDIX B: GAMS CODE FOR MIP MODEL

This is an example for a model with 15 defender strategies and 15 attacker options.

```
sets
k payoff branches /1*9/
d defender strategies /1*32768/
a attack scenarios /1*15/
alias (a,h);
alias (d,dd);

parameter c(d)
/
$ondelim
$include stratcosts-15by15.txt
$offdelim
/;
*display c;

table p(d,a)
$ondelim
$include pijfile-15by15.txt
$offdelim
;
*display p;

parameter py(k)
/
$ondelim
$include py_k.csv
$offdelim
/;
*display py;

table yield(k,a)
$ondelim
$include yields15.csv
$offdelim
;
*display yield;

scalar M big M /10000/;
scalar failedattack cost of a failed attack to attacker /0/;

parameter EVY(a)
        delta(d,a)
        omega(d,a);
EVY(a) = sum(k, py(k)*yield(k,a));
delta(d,a) = EVY(a)*p(d,a)+c(d);
```

```
omega(d,a) = EVY(a)*p(d,a)+(1-p(d,a))*failedattack;

variables
beta     bound variable for attacker payoff
y(d,a)   joint strategy variable (1 if d and a chosen)
alpha(a)       attacker choice variable (1 if scenario a chosen)
OBJ      objective expected value;
binary variable alpha,y;

EQUATIONS
  dOBJ           define actual objective function
  ONEJOINT       only one joint strategy can be chosen
  JOINTATTACK1(a) if alpha(a)=1 then some y(d 1) must be 1
  JOINTATTACK2(a) if alpha(a)=1 then some y(d 1) must be 1
  ONEATTACK      only one attack scenario can be chosen
  ENFORCEATTACKPAY1(a) makes sure the attacker's payoff is considered
  ENFORCEATTACKPAY2(a) makes sure the attacker's payoff is considered;

  dOBJ..         OBJ =e= sum((d,a),delta(d,a)*y(d,a));
  ONEJOINT..     sum((d,a),y(d,a))=e=1;
  JOINTATTACK1(a)..  alpha(a) =l= sum(d,y(d,a));
  JOINTATTACK2(a)..  sum(d,y(d,a)) =l= 1;
  ONEATTACK..     sum(a,alpha(a))=e=1;
  ENFORCEATTACKPAY1(a).. 0=l=(beta-(sum(d,omega(d,a)*sum(h,y(d,h)))));
  ENFORCEATTACKPAY2(a).. beta-sum(d,omega(d,a)*sum(h,y(d,h)))=l=(1-alpha(a))*M;

  OPTION Reslim = 5000;

MODEL UREX /ALL/ ;
SOLVE UREX USING MIP MINIMIZING OBJ ;
DISPLAY y.l,beta.l;
```