A thesis submitted for the degree of Master of Philosophy

INTEGRATING SYSTEMS AND ECONOMIC MODELS FOR SECURITY INVESTMENTS IN THE PRESENCE OF DYNAMIC STOCHASTIC SHOCKS

Hasiba Afzalzada

Declaration

I, Hasiba Afzalzada, confirm that the work presented in this thesis, "Integrating Systems and Economic Models for Security Investments in the Presence of Dynamic Stochastic Shocks ", is the result of my research carried out under the supervision of Professor David Pym. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Acknowledgement

First and foremost, I want to take this time to thank Allah (SWT) for giving me the opportunity to carry out this research and allow me to finish writing this thesis today.

Alhamdulillah.

My heartfelt gratitude goes towards my supervisor, Professor David Pym, who, despite his busy schedules, offered me invaluable and immeasurable guidance, advice and constructive criticisms over the course of my studies which allowed me to carry out this research.

I would also like to thank Professor Christos Ioannidis from Aston University, for his input and advice throughout my studies and Professor Julian Williams from Durham and Dr. Tristan Caulfield from UCL for the help they provided me when setting up the economic model in MATLAB.

A special thanks to all my family members for their continued encouragement and support throughout these years, in particular my mum and mother-in-law for taking care of my son whilst I finished off writing this thesis, and my dad who always taught me to work hard and aim high, but also supported me immensely when things didn't go as planned in the past few years.

Last, but certainly not least, I would like to thank my husband who has been incredibly supportive, encouraging and patient with me throughout this rollercoaster-like period of my life and who believed in me when I did not. I love you.

Abstract

Organizations deploy a number of security measures with differing intensities to protect their company's information assets. These assets are found in various location within a company, with differing levels of security applied to them. Such measures protect the different aspects of the organization's information systems, which are typically separated into three different attributes; confidentiality, integrity, and availability. We start by defining a system in terms of its locations, resources and processes to use as an underlying framework for our security model. We then systematically define the time evolution of all the three attributes when subjected to shocks aiming at degrading the system's capacity. We shock each of the attributes of the system and trace the adjustment of the attributes and policy responses; we undertake this exercise for different types of organizations: a military weapons system operator, a financial firm or bank, a retail organization, and a medical research organization, producing their impulse-response functions to quantify their responses and speed of adjustment. This economic model is validated through various means, including Monte Carlo simulations. We find that organizations, although they react in similar ways to shocks to their attributes over time, and are able quickly to get back to their pre-shock states over time, differ in the intensity of their policy responses which differ depending upon the character of the organization.

Impact Statement

Security is a widely researched area, which can be separated into many different areas, such as cybersecurity, information security, threat protection or economic security, to name a few. Each of these areas have their own models which aid in solving their management problems. The integration of models from separate areas of research is an area that has not yet been covered, and is an area I have aimed to contribute to in this thesis.

The main contribution of this thesis is the integration of two theoretical areas, namely, economic models of security investments and distributed systems security. This allows the more precise modelling of security management.

The idea is that integrations such as these could be used to support research into more accurate analyses of real-life problems. For example, how economic models can be used to inform choices about security designs or how distributed systems models can be used to understand a variety of systems security problems.

Outside of academia, this work could be the beginning of decision-support tools for people who are engaged in the management of security investments. As more security data is obtained from companies, this model could be improved and refined to better represent real-life scenarios. For example,

- it could aid in the decision to invest in the right security assets to obtain the desired level of security for a specific company,
- it could allow companies with varying preferences in their security attributes to see the effects of changes in particular investments to their levels of security, or
- it could also provide them to with ways to recover from certain attacks, over time.

With each improvement, this model would get closer to accurately modelling real security systems and so could help the companies to achieve a balance between the level of security maintained and the level of investment required that is appropriate for their business.

As more companies become willing to explore modelling techniques such as these – and so expose some of their experience and data – the accuracy and usefulness of this model can be expected to improve.

Contents

List of Tables

List of Figures

1 Introduction

Business, computer, social and economic networks all depend on some degree of security for their operation. A small store, for example, may have a CCTV unit to monitor customer behaviour whilst a typical office environment may have several policies and procedures, which must be adhered to, in order to keep it secure. Each of these security measures would have different aims and efficiencies to prevent any loss or damage to the organisation.

The use of different types of security measures at various organisations provides an indication to a security manager's preferences in protecting various assets. For example, some may prioritise the protection of the data from getting accessed by unauthorised users, whereas others may be more concerned about sufficient availability of products to their customers. These preferences are classically separated into three different attributes: confidentiality, integrity and availability. As a quick overview, confidentiality refers to the prevention of sensitive information reaching malicious users while ensuring that the right people are still able to access it, integrity involves the safekeeping of data accuracy and availability refers to the system's property that enables reliable and predictable access to information by authorised agents. A breach in any of these could result in serious consequences for the organisation. More detailed explanation of each of these attributes along with ways to maintain them is given in Section 4.

Attackers also have different preferences when attacking an organisation; for example, a virus may want to affect the performance of a computer to gain access to sensitive data, such as the recent NHS WannaCrypt ransomware outbreak [173, 130, 58]. This would affect the availability and confidentiality of the system. On the other hand, a burglar may be able to gain access to several assets and damage them once he gets inside a building, which would affect their integrity. The preferences of the attackers, therefore, would also need to be taken into consideration when trying to predict its movements in an organisation. This aids in predicting the expected damages it could cause to the organisation.

Much research has been conducted regarding individual parts of these problems, (a detailed analysis of which is provided in Section 2) such as the relationships between attackers and defenders in security games, or the amount of investments required to be made in specific organisations. However, the integration of these different ideas has not been covered in the literature, which is an area we would like to contribute to.

1.1 Thesis Contents

The aim of his thesis will be to provide an integration of different concepts taken from three separate areas of research; Distributed systems, the Economics of Information Security and Stochastic Security Games to help in understanding the following:

1. How an organisation's security network can be represented in the distributed systems framework and used as an underlying network.
2. The behaviour of organisation with different preferences when attacked and the investments they are required to make to overcome these.
3. The movement of attackers in a secure system and their interaction with various types of defences placed at different locations.
4. The expected damages caused by an attacker in a system with interdependent defensive measures placed at different locations.

To tackle these questions, we first explore different literatures separately to understand the individual concepts, after which we integrate some of the concepts to form a mathematical model that is able to capture the information required to answer the proposed questions.

The thesis starts by introducing the concept of distributed systems in terms of locations, resources, processes and environment to have a solid representation of the security system's network and the assets that are being protected. Each of the mentioned terms are defined and captured mathematically using a combination of graph-like structures (to capture the different locations), process calculus, (to represent resources) and probability distributions (to model the environments), [53]. Different examples of the applicability of this concept to real systems is given and explained to the reader in Section 3.

An economic security model, in terms of the trade-offs between the three security attributes confidentiality, integrity and availability as well as the investments made by an organisation is then set up based on [94]. This aids in understanding the behaviour of different organisations when attacked. We explain the trade-offs between each of the attributes and investments with a given set of assumptions and we use loss functions and impulse-response functions to analyse the behaviour of an organisation, satisfying the given assumptions, when attacked and how it returns back to its equilibrium state.

In this economic model, we represent the complexity of the system simply as a function of the level of interconnectivity between locations, but do not explicitly show how they are interconnected. We understand that this is a limitation of our model and, therefore, enrich the model by embedding the distributed systems framework within it. This will allow for the movement of the attacker to be captured and the damages it causes to the system's locations to be observed closely.

Here, the properties of each of the resources are modelled in terms of the security attributes, rather than their explicit mathematical representation and each of the locations in the system are embedded within the matrix representations of the security attributes and investment.

In order to develop a model for an attacker, we propose to use a Markov Decision Process to create a better illustration of the propagation of an attacker through a network. This can be used as an underlying framework in a two-player, zero-sum Markov Security game (as in [9, 10, 22]) to obtain the optimal strategies for the attacker and defender in the security system as well as their expected pay offs/losses produced, respectively. In future work, this model can be integrated with the Enriched Economic Security model (as in Section 6) to provide a more accurate model of the behaviour of attackers in a system.

A general overview of this thesis is illustrated in Figure 1.1.

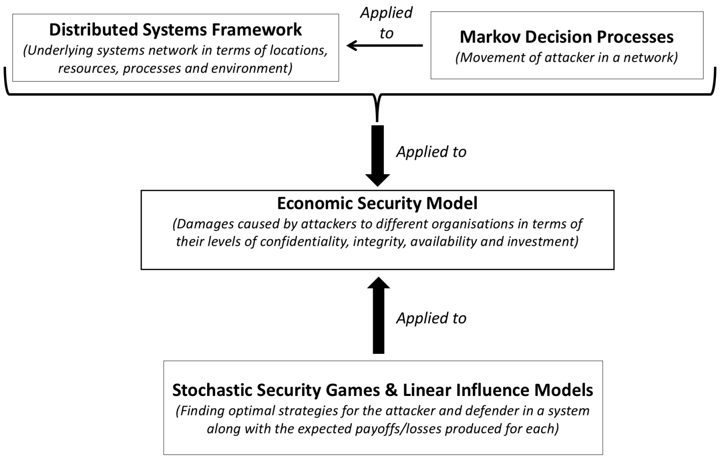

Distributed Systems Framework
(Underlying systems network in terms of locations, resources, processes and environment)
Applied to
Markov Decision Processes
(Movement of attacker in a network)
Applied to
Economic Security Model
(Damages caused by attackers to different organisations in terms of their levels of confidentiality, integrity, availability and investment)
Applied to
Stochastic Security Games & Linear Influence Models
(Finding optimal strategies for the attacker and defender in a system along with the expected payoffs/losses produced for each)

Figure 1.1: Thesis overview illustrating the contributions to be made.

1.2 Mathematical Modelling Methodology

In order to deliver these models, we use a mathematical modelling methodology to create a valid mathematical formulation of our problem in order to better understand the situation and accurately predict its future behaviour. A general form of the mathematical modelling cycle is given in Figure 1.2, which splits the process into 5 different stages: Real World Problem, Working Model, Mathematical Model, Computational Model and Conclusions and Results, [8].

As a quick overview of this cycle, the initial stage involves identifying a "Real World Problem" and performing appropriate background research to help focus on a workable problem. This problem is then simplified to form a "Working Model" where the important aspects of the problem are identified and any unnecessary information is eliminated. A set of assumptions is also made and the relationship between useful model variables are identified within the context of the assumptions. This model is then represented in a "Mathematical Model" where the solutions to the mathematical equations used describe the solutions to the "Working Model". The "Mathematical Model" is then translated into a "Computational Model" to obtain solutions to the model that has been set up. If the model is simple enough, it may be solved analytically. Otherwise, a computer program will be required such as MATLAB, Excel, or Julia. The computer code is then executed to obtain results, explore the ranges of parameters for which this model is valid, and formulate conclusions. The "Conclusions and Results" are then interpreted and compared with the "Real World Problem" behaviour. If the model results do not agree with the physical or experimental data, the "Working Model" will have to be re-examined (possibly by relaxing assumptions or editing variable relationships) and the consequent modelling steps will have to be repeated. Modelling processes usually proceed through several iterations until the model is "acceptable" to the modeller according to their judgement against observed real world behaviour.

Our models are developed using this mathematical modelling cycle. We begin by carrying out some background research in Section 2 to help us in understanding the problems that have been dealt with and any remaining gaps in the literature. We concentrate on one gap in the literature out of the many available and start to develop a working model for it.

Section 4 develops a new working model based on [94]. Here, its mathematical model is improved through the addition of integrity into the equations and the trade-offs between the security attributes are set based on a set of assumptions as determined in the working model. The stability conditions for the model are also

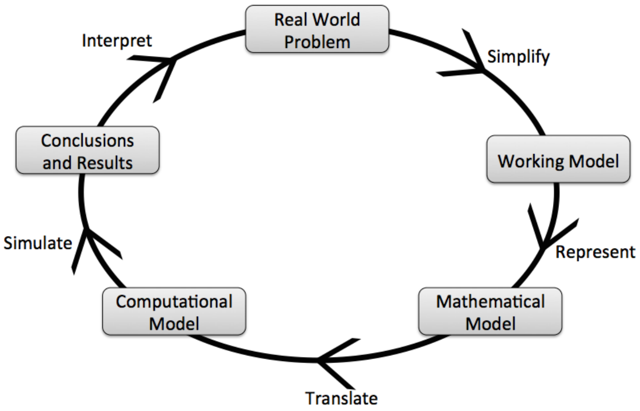


Figure 1.2: Mathematical Modelling Cycle

computed and a sensitivity analysis is performed to further validate the model.

This mathematical model is then computed into MATLAB and simulations are performed to gain an understanding of the behaviour of organisations with different security preferences when attacked by the same attacker. The results are analysed and conclusions are made regarding the accuracy of the model and any discrepancies between the model and real world behaviour are taken care of.

This cycle is repeated in Section 6 where the model now incorporates the distributed systems framework. Changes to the mathematical and computational model are made as required and conclusions are established based on the results.

The validity of our model is ensured by following the common mathematical modelling cycle which ensures that the representation of the problem and model's structure, logic and mathematical causal relationships are reasonable for their intended purposes. We establish its validity and robustness by carrying out a sensitivity analysis, using Monte-Carlo simulations, and stability analysis, to ensure that the relationships determined in the model provide a good representation of the real world system for a range of parameter and variable values. In addition to this, we validate our model by setting up a number of thought experiments. These will aid in testing the integrity of the set up of the model to ensure that it makes sense and can be used. 4 different organisations are considered to explore the sensitivity of the model to particular parameter choices and to check whether the results reflect the behaviour of similar organisations in real life.

We believe this model can help us to understand how to best invest in security measures to obtain the optimal level of security for a specific organisation, through the integration of the different concepts taken from various fields. In particular, this model will allow for the understanding of the behaviour of an organisation when specific locations are being attacked.

1.3 Summary of Contributions

The contributions made in this thesis involve the integration of individual concepts from various literatures and the set up of an economic security model to understand how organisations with different security preferences behave when attacked. A brief summary of the contributions made is given below.

1. Set up of an Economic Security Model which captures the trade-offs between the three security attributes and investment and shows the behaviour of organisations in the presence of dynamic, stochastic shocks (Section 4).

2. Enrichment of the Economic Security Model with the application of the Distributed Systems Framework to obtain a more accurate representation of the complexity of a system in terms of its locations and resources (Section 6).

3. Integration of Markov Decision Processes with the Distributed Systems Framework to show how an attacker would move through a network in terms of locations, resources and processes (Section 7.4).

4. Integration of the previous contribution with Stochastic Security Games involving Linear Influence Model to obtain a model for the attacker's movement in a system which contains interdependent defensive measures at different locations to protect the network, as well as the expected damages that could be caused by the attacker (Section 7.7).

In the following sections, we start by providing a comprehensive literature review of the different areas of research mentioned earlier in Section 2. We then move on to introducing the concept of distributed systems and the mathematical representations of each of its components along with some examples of different real life systems in the distributed systems framework in Section 3. After this, we explain the development of the initial economic security model in Section 4, which is then enriched with the distributed systems framework in Section 6.

Following this, we explore some types of real life attacks and explain the notion of Markov decision processes and Markov Security games. We show that these ideas can be used as a way to modelling attackers and introduce the concept of Linear

Influence Models in Section 7. In this section, we also provide a way to solve the Markov Security games in a zero-sum game which is integrated with the linear influence model, as in [9, 10].

Finally, Section 8 discusses conclusions and provides recommendations for the extension of this model for future work.

Due to the use of many different parameter notations in each of the sections, a table has been compiled in the Appendix which summarises each of their representation for the reader's convenience.

2 Literature Review

The security literature which captures the context of this thesis can conveniently be divided into 3 different areas: Networks, Attack models, and Security Economics.

In what follows, we give a brief overview of some of the work carried out in the security area in general, ranging from its usability to some previously arisen privacy concerns. We will then concentrate on 3 main aspects of security, namely, networks and the different ways it has been represented previously and attackers and the different ways they could be modelled through, for example, (Partially Observable) Markov Decision Processes or Hawkes Processes. We will also briefly discuss the modelling of interactions between attackers and defenders in networks as part of a Security Game. Finally, we will explore the security economics literature which considers how any damages caused to the organisation by attackers could affect its financial situation. These could be the losses that an organisation faces after attack, or the investments to be made to prevent such attacks in the future.

2.1 Security

Security entails the protection of assets from various threats posed by certain vulnerabilities, where its strength is described as a 'chain which is as secure as its weakest link' by Schneier (2004).

Research in security has received increased attention over the past few decades regarding, for example, its usability in organisations, encryption, anonymity & privacy, intrusion detection systems, and issues in cyber security, amongst many others, [116, 43, 148, 42].

The terms 'cyber security' and 'information security' are often used interchangeably to describe these types of protection of assets from possible harm. Von Solms and van Niekerk (2013), however, argue that, although there is a substantial overlap between cyber and information security, these two concepts are not completely analogous. They argue that information security only refers to the protection of information and its crucial elements, in terms of their confidentiality, integrity and availability. This includes the systems and hardware that use, store, and transmit that information, [179, 97]. Cybersecurity on the other hand, also encompasses the protection of ethical issues and humans and their interest, such as cyberbullying or the control of common household appliances through the internet, in addition to the preservation of information and assets against relevant security risks in the cyber environment.

We aim to develop a model that is able to capture the protection of information in an organisation, as well as its assets, from both physical and cyber attacks, in terms of their levels of protection of the confidentiality, integrity and availability.

Organisations have started to recognise that the safeguarding of their assets and data require more than the deployment of physical and technical controls; human behaviour, for example, must also be taken into consideration as a potential source of vulnerability, such as user compliance to security policies or the usability of security products.

Caputo et al. (2016) explore three organisational attempts to provide usable security products to their employees. They find that improved usability does not result from efforts from a single individual who cares about usable security nor from the team's prior experience in building usable security, but from companies that are motivated to improve the usability only if it is clear that this will decrease the negative consequences of the usability problems. That is, useable security is particularly important to organisations if it means an increase in sales or increased compliance with security policies.

Many organisations have started to deploy a set of security policies to protect their assets, which consists of a collection of principles and rules that describe how an organisation intends to protect the three main security attributes of its system. Much research has been carried out in the safekeeping of each of these attributes, and although other characterisations of security exist, such as criticality and sensitivity levels or accuracy, authenticity, utility and possesion, [179], the CIA Triad remains a common way to group them.

Prasad et al. (2011), for example, present a framework which focuses on data leakage in cloud computing by categorising the data into the three security attributes and finding a three dimensional approach to authenticate their data. They state that their model provides availability of data by overcoming many existing problems such as denial of services or data leakage as well as providing more flexibility and capability to meet the new demand of today's complex and diverse network.

Olivier (2002), on the other hand, explores the challenges of database privacy, by finding a balance between the three security attributes, to enable the storage of personal information in databases. He notes that this balance should not necessarily be a trade-off but should take into account the sensitive nature of the data being stored and attempts to increase all three dimensions to the highest level possible.

We aim to develop a model which is also able to capture the the protection of assets in organisations in terms of the three security attributes.

Privacy is another area in the security literature which has received much attention, [129, 148, 113, 85, 140, 63] to, for example, understand the optimal balance between a user sharing his data when using a service whilst also maintaining a level of privacy they prefer.

Shokri (2014), for example, explores this balance by proposing a methodology to minimise the utility cost of obfuscation whilst also guaranteeing the user's desired level of privacy to the ultimate extent that is theoretically possible. He designs obfuscation mechanisms that provably limit the user's privacy risk against any inference attack and any information leakage through observation.

Caulfield et al. (2016), characterise privacy based on four key factors: context, which is the setting and purpose for which a given technology is used, requirement, which refers to the level of privacy a technology must provide for an agent to be willing to use it, belief, which is the agent's perception of the level of privacy provided by the technology and the relative value of privacy, which refers to how much an agent cares about privacy in this context.

These concepts are introduced into their model to capture the variations in requirement, belief and relative value in the population. Their model indicates expected levels of adoption of the competing technologies in different contexts, such as sending content with varying degrees of sensitivity over a service such as Snapchat.

Many other applications of privacy have been also been explored, such as its involvement in the use of serious games in different organisations, [117], or its protection disclosing private data to researchers [166].

The incorporation of the representation of user privacy could be an element to be considered in our model. The notion of privacy of data could be argued to be similar to the preservation of its confidentiality, and therefore could possibly be captured implicitly in our model.

Different defense mechanisms to help protect an organisation have been developed in the literature, some of which will be discussed in Section 2.3. However, in order to understand how to protect an organisation from attack, we must first understand its security system's architecture. Each organisation's security architectures differs in the way that they have been set up and the assets they aim to protect. In the next section, we explore some of the ways these architectures can be represented in the form of network models to find an effective way to represent any security architecture in our model.

2.2 Networks

Network models have traditionally been used to represent a variety of different scenarios in varying disciplines such as particle physics, finance, biology, economics, and many others. They have been used to model, for example, the spread of infectious diseases amongst populations [32, 188], the structure of relationships between social entities, [70, 114], economic interactions in financial networks, [7] or the different aspects in informations systems architectures, [82].

Although models from different disciplines could be used to model various aspects of attackers in a system, such as the diffusion model developed in Bonaccorsi et al. (2004) to describe the rate of spread of attacks in a system, we will limit our research to network representations of systems architectures and their security properties only.

We start by exploring the classical distributed systems framework as described in, for example, [53, 55], where models are based on concepts of locations, resources and processes. Locations are the places within a system where resources reside, and resources are the components of the system which can be manipulated by processes. Processes are the concept that describe the dynamics of the system, which manipulate resources in order to deliver services. Further details of these and their mathematical representations are given in Section 3 to use as an initial underlying framework which describes an organisation's security architecture.

This concept has also been used as an underlying framework in, for example, Caulfield and Pym (2015) to aid in the development of a rigorous modelling framework that can be used to help security managers make better decision in designing security policies that deliver objectives required by organisations. Here, complex models are expressed by smaller, complete models that are represented in terms of locations, resources and processes. These are then composed together using the notion of interfaces to form the original complex system. However, one of the limitations of this model includes the absence of the incorporation of psychological factors which could influence an agent's decision.

Another way to analyse issues related to systems has been through the use of directed graphs which consists of a set of vertices and edges where each of the vertices represent different assets in a system and any links between them are given by the edges.

McCarthy et al. (2016), for example, have used directed graphs to describe a local computer network, where each of the vertices correspond to the set of hosts in the network and each of the edges show that communication between hosts is allowed. They use this representation, with the use of POMDPs, to address advanced persistent threats (APTs) in a computer network. In particular, they focus on detecting data exfiltration over DNS queries where any existing detection sensors are imperfect and lead to noisy observations about the network's security state.

We will use similar concepts to develop an attack model in Section 7.5, which is able to capture the movement of an attacker in a system as well as the damages that it causes. Since an attack on one of the locations in a system would affect the level of vulnerability of the next, we would require some type of influence models to capture this change.

Bagchi et al. (2016) have used interdependency graphs to represent the assets in a networked (cyber-physical) system as vertices on a directed graph. Here, the presence of an edge between two vertices indicates that if one of the vertices has been compromised, it can be used to launch an attack on its connected vertex with a given probability.

Linear Influence networks have also recently been used to model interdependencies between vertices in a network. These could be influences of security assets from one location to another, as described in [9, 10], or the influence of vulnerabilities on each other when one of them has been attacked, [10, 22]. Here, weighted directed graphs are used to determine the level of influence between vertices.

Bambos et al. (2016), for example, uses this notion to capture inter-agent influences on a network. That is, an agent's actions can directly affect his interacting neighbours either in a positive or negative way. Therefore, the neighbour must take into account the consequences of such influence before making his own action.

Although not all influences amongst vertices or agents are bound to be linear, as an initial starting point we will only consider linear influences which can be extended to non-linear cases in future work. Further details regarding the set up of linear influence models and how to solve them are given in Section 7.6.

We aim to use a combination of the ideas presented above to obtain a solid underlying framework of a system's security network. That is, a systems network model that is able to effectively capture the resources to be protected, the defensive measures used to protect them, and the movements and effects an attacker has on different parts of a system.

One of the challenges of this work includes finding an accurate representation of an attacker, in an interdependent system, which can be integrated into our underlying framework of security systems. Attack models have been presented in numerous ways in the literature and differ quite significantly in the way that they have been set up and the events that they model.

The next section explores the different ways attackers have been represented in the literature over the years, some parts of which will be incorporated in our model.

2.3 Attacker Models

Different representation of attacks and attackers in systems can be found, ranging from (Partially observable) Markov Decision Processes to determine the defender's or attackers' decisions at each point in time, to attack trees which systematically categorise the possible ways in which a system can be attacked, [41, 119, 74, 111, 124, 134, 171].

Spreading mechanisms have also been used to describe attackers' behaviour, such as in Zhang et al. (2015), where they are used to describe the spreading of computer worms Conficker and Code Red. Here, they start by defining two ways an epidemic can spread itself in a system; local spreading, in which infected vertices can only infect a limited set of directed target vertices, and global spreading, where an infected vertex can infect any other vertex. In reality, many epidemics use a hybrid mixture of both types of spreading. A mathematical framework is proposed to study hybrid epidemics and focuses on exploring the optimum balance between local and global spreading in order to maximize outbreak size.

They find that hybrid epidemics can cause larger outbreaks in metapopulations, which consist of a number of subpopulations (collection of densely or strongly connected vertices) than a single spreading mechanism. These results could be used to manipulate the balance between local and global spreading in order to provide a way to estimate the largest outbreak of a hybrid epidemic which could pose serious threats to internet security.

This spreading mechanism only provides a way of estimating the size of the outbreak and is only limited to attackers which randomly move around in a network. For our model, we are interested in the way an attacker moves in a system, which this mechanism fails to provide us. In addition to this, we are also interested in the decisions it makes at each point in time to find its optimal path in reaching its target, given the defensive measures present.

Another way to represent attacks is through attack graphs, which are restricted to directed acyclic graphs by employing a monotonicity assumption on the attacker's behaviour, as in [12, 60]. That is, the attacker gains increasing control of the network as time progresses, never willingly giving up previous attainments.

Miehling et al. (2015), consider the movement of attackers in a system using Bayesian attack graphs and Partially Observable Markov Decision Processes (POMDPs).

In the Bayesian attack graph, each of the vertices represent attacker capabilities, such as vulnerabilities of a service, or information leakage, and the edges are exploits, which are events that allow the attacker to use their current set of capabilities to obtain additional capabilities.

The attacker's movement in the graph is modelled by a probabilistic spreading process, where it is, again, assumed that it moves randomly throughout the network with no specific, intelligent process, and the defender's decisions are calculated using Partially Observable Markov Decision Processes. A small network is set up to obtain an optimal policy which maps the current belief to the optimal countermeasure action.

These attack graphs can be interpreted into Markov Decision Models, as in Jha et al. (2002), where they have been used to determine the optimal defence strategies that minimise the probability of successful attack.

(PO)MDPs have commonly been used to calculate defence strategies in systems in the presence of attacks, [185, 111, 4, 86].

Wu et al. (2012), for example, use MDPs to derive an optimal strategy for defending against jamming attacks in cognitive radio networks, where several attackers intend to jam the unlicensed user's communication link by injecting interference. They consider situations where unlicensed users could hop across multiple bands to avoid being jammed. They derive the optimal defence strategy which balances the cost associated with hopping and the damage caused by the attackers.

It is recognised here that the users may not know some information, such as the number of attacks. Therefore, a learning process is introduced where the user uses past observations using maximum likelihood estimation and estimates the required parameters for the MDP accordingly. They find that the attackers should adopt a strategy that randomly scans all the bands to find the users so that the optimal defence strategy can be obtained from the value iteration of the MDP.

MDPs have also been used to model the interactions between honeypot and botmasters in Hayatle et al. (2013). Honeypot operators need to choose the optimal response that balances between being disclosed and being liable for participating in illicit actions. This model is then extended to POMDPs to model the uncertainty of the honeypot state by the botmaster. They find that exploiting the legal liability of honeypots allows botmasters to have the upper hand in their conflict with them. They also show simulation results which show the optimal response strategies for honey pots and their expected rewards under different attack scenarios.

Caulfield and Fielder (2015) use POMDPs to model computer networks and vulnerabilities that can be used to find the optimal allocation of time to different system defence tasks.

Here, the defender is taken to be the system administrator who is thought to have the following actions: monitoring the network, patching vulnerabilities, recovering the system after an attack, or carrying out other tasks related to maintaining the system. The attacker is taken to have 2 different actions; using exploits to advance through a system towards their target by only using a single attack at a time, or take no action, for example when they do not want to use an exploit or have no remaining usable exploits. A security game is set up which describes the network, vulnerabilities, players and costs and the solution is given as a policy which indicates the optimal actions to take for the defender at a given state.

They find that system administrators must spend on essential security-related tasks only and spend the majority of their time on non-security tasks.

In our model, however, we concentrate on the attacker's movement throughout a network and assume that he has full knowledge of the system that he is going to attack.Therefore, rather than modelling the defence strategies, we aim to use MDPS to model the movement of an attacker in a system in which the attacker decides on the optimal action to take (which location to attack next), given the set of states available (set of directly connected locations), their respective transition probabilities (probability of successful attack), and associated rewards (how much damage it will cause whilst also getting closer to its target).

However, since the relationships between locations in our system are dependent on each other, and such relationships are not taken into consideration in MDPs, we would need to extend our model to incorporate this.

Inter-dependent relationships between threats in a system have been recognised previously in the work carried out by Baldwin et al. (2012), where they develop a model which captures infrequent inter-relationships between threats as well as the manager's responses for given temporal relationships between the number of attacks to the system, their change (or 'jump') in frequency, and the extent of its impact. They characterise the operational status of their systems in terms of their levels of criticality and sensitivity, where the former refers to the importance of the availability of accurate information for continuing system operations, and the latter is concerned with the level of security required for protecting data from access by unauthorised agents.

They use Hawkes Processes, a model of contagion, to assess the existence of contagious behaviour between threats to critical services, such as email, databases and website operation, using threat data by DShield. They find that attacks on individual ports are, indeed, inter-related, with the relationships being exposed by the estimation of jumps and mutually self-exciting behaviour.

Although this model has been useful in determining the existence of interdependent relationships between threats in a system, it does not show the effect of these threats on the interdependence between locations in a system. Therefore, we will need to consider a different approach to capture this effect in our model.

Influence models have also been used to capture interdependencies between vertices in networks. As introduced by Miura-Ko et al., (2008), and further developed by Alpcan et al. (2009) and Alpcan and Başar (2011), these models capture the influence of vulnerabilities at different locations in systems as well as the influences of various security controls on each other. A stochastic security game has been set up in Alpcan et al. (2009) which models the interactions between the attacker and defender in a given security system. These ideas will be further discussed and applied to our model in Sections 7.5 - 7.7.

The concepts from game theory have received much attention in the past two decades to model various situations, [24, 98, 123, 9, 22, 10]. Here, the optimal actions of both the attackers and defenders are established in diverse system architectures which consist of agents, connected by physical or virtual links, who must decide on an action, given the actions of the other users and the network structure.

Game theory has been applied to many different fields, including social, economic, security or financial networks ([7, 28, 66]). The theory has been developed for small scale, sophisticated interactions which is based on strong assumptions such as common knowledge and forward-looking behaviour, [138].

Researchers have explored the applicability of game theoretic approaches in many security and privacy-related computer and communication networks. In particular, network security mechanisms in a game theoretical field have received immense attention from the research community.

Ellis et al. (2010) provide a neat representation of existing game theoretic solutions which are designed to enhance networks security and present a taxonomy for classifying the proposed solutions. They point out that although many types of games exist, such as static game models or games with complete or perfect information, in reality, a network administrator often faces a dynamic game with incomplete and imperfect information against an attacker. Some work has been carried out involving incomplete and imperfect information specific to wireless networks, however, much more research is required in many other security situations.

A structured and comprehensive review of a selected set of works has been given in Manshaei et al., (2011), in which they are categorised in 6 sections: security of the physical and MAC layers, security of self-organising networks, intrusion detection systems, anonymity and privacy, economics of network security and cryptography.

Some common concepts applied to these security problems include Stackelberg security games [49, 109], Nash equilibrium [131, 18, 28], (non)zero-sum games [104, 189, 35], and stochastic games [192, 190, 26, 165, 84].

A Stackelberg security game consists of two players, and a possible set of targets. Here, one of the players is the leader (defender), who can decide upon randomised policy of defending the targets, and the other the follower (attacker), who is assumed to observe the policy of the leader upon which it chooses a target so as to maximise its expected utility.

A Nash equilibrium, as introduced by Nash in 1950, is a collection of strategies for each of the players such that each player's strategy is a best-response to the other players' strategies and each of the decisions are made simultaneously by each of the players. This means that no player can get a higher payoff by changing strategies given that the other players also don't change strategies.

The concept of equilibria has been proven to be applicable to zero-sum games by Shapley (1953) and to nonzero-sum stochastic games by Fink (1964).

Zero-sum games are mathematical representations of a situation in which a player's gain or loss is exactly balanced by the losses or gains of the other players. That is, the sum of the total gains and losses of all of the players is equal to zero. These are often solved with the minimax theorem, or Nash equilibrium. On the other hand, the total gains and losses of all the players in a nonzero-sum game can be less than or more than zero, [183]. An overview of the many zero-sum games, which includes all basic streams of research in this area such as vector payoffs, incomplete information and algorithms amongst others can be found in Jaśkiewicz and Nowak (2016) for the interested reader.

Nonzero-sum stochastic games pose a serious challenge in terms of convergence of solutions due to the non-uniqueness of Nash equilibrium at each state. Although many methods have been suggested to overcome these problems, there is still no unified theory that is easily applicable to solve nonzero-sum stochastic security games, [9]. For a more detailed overview of all basic streams of research in the area of nonzero-sum stochastic games, such as algorithms, stopping games and correlated and uniform equilibria, amongst others, the reader is referred to [100].

In our model, we use MDPs as an underlying process to describe the decisions made by an attacker in a zero-sum stochastic security game. We find the optimal payoff of an attacker in a zero-sum game which captures the influences of vertices on each other in a network. Further details regarding the set up of this game are given in Section 7.5.

It is worth noting that not all 'attacks' are a deliberate act of sabotage. Some are also due entirely to inadvertent errors, such as natural disasters or even the accidental loss of important data (through, for example, spilling drinks), as pointed out by Vorobeychick and Letchford (2012). Although such failures are generally far more common than attacks, the vast majority of work in security games posits an attacker, but ignores such failures entirely. One paper which does take this into consideration is by Zhuang and Bier (2007), which uses a Nash equilibrium to set up a rigorous model for balancing defence against terrorism and natural disasters. They find that increased defensive investments indicate that an attacker can either increase or decrease his level of effort, to help compensate for the reduced probability of damage from an attack, or because attacking is less profitable at high levels of defensive investment, respectively.

We leave the incorporation of accidental damage to the system as an additional extension to our model in future work.

We now move on to exploring the security economics literature to understand how to best invest in security measures in the following section.

2.4 Security Economics

The security economics literature has become a fast-moving and thriving area of research in recent years, [13]. As systems are becoming more advanced, incentives are becoming as important to dependability as technical design. Economic theory and models are used to analyse incentives between the involved stakeholders. Cavusoglu (2004) argues that information security should be viewed as a value creator that supports and enables e-business operations, rather than just an expense. He claims that this value can be created by developing secure environments for information and transaction flows between companies and their partners.

Because of the large nature of this literature, we will only concentrate on economic security research which incorporate the presence of shocks or attackers into their model and the investments that should be made to protect organisations. The reader interested in the various other aspects of this literature is referred to [13, 158, 87, 152] as a starting point to obtain an initial insight into some of the other works carried out in this field.

The concept of attacker behaviour has been modelled in various ways in different scenarios, some of which concentrate on the spread of attack within a single organisation, others which observe the effects of attacks on a number of interconnected organisations, [68, 56, 103, 121, 136]. Our model aims to be able to define a system's network in such a way that it is able to capture both a single organisation's network and a network of connections between various organisations.

Cremonini and Nizovtsev (2006) consider an economic model of behaviour of attackers in two cases; one in which attackers are able to obtain information about each target's security levels and the other when they are not. They find that, in the first case, the attacker's optimal strategy is to attack systems with low security levels more than those with higher security levels. An increase in the defender's security level then affects the frequency of security incidents in two ways; direct and indirect.

The direct effect refers to the technical characteristics of a system and decreases the probability of successful attack. The indirect effect relates to the amount of effort the attacker puts into attacking the system, which further decreases the frequency of security incidents. Although the indirect effect is usually overlooked, its effect could greatly exceed that of the direct one.

In the second case, there is no indirect effect on the system, and the attackers treat every target the same. In this scenario, it has been found that systems with low security levels compared to the rest of the population should try to keep their security measures a secret, whereas those with higher security levels are advised to reveal some of their security measures to discourage attackers from attacking them.

Although this study has obtained some interesting results, its static nature limits its applicability to the real world. The interrelationships between the attacker and defender should be represented in a dynamic, game theoretic model to be able to observe more efficient results. The use of game theoretic models in the security economic literature will be discussed later in the section.

Acemoglu et al. (2015) have also developed a model which captures the propagation of shocks to different, interconnected organisations. Here, a model is developed to determine the 'downstream propagation of supply-side shocks' and 'upstream propagation of demand side shocks' to understand the effects of a shock to different sectors in the economy. The Cobb-Doughlas production function is used, in the presence of a shock, to determine the amount of output that can be produced through 2 or more inputs (e.g. capital and labour) and to obtain the Leontief inverse of an input-output matrix. That is, the final demand of each organisation for the given input-output matrix. This matrix shows how the output from one sector could become an input to another sector. In particular, each column of the matrix represents (monetary) values of inputs to each sector and each row represents the (monetary) value of each sector's outputs. The propagation of shocks can then be observed through this model.

This idea could be used to understand the spread of an attack in a connected system, in terms of its monetary values, by specifying the labour and capital inputs as well as the value of shocks.

The damage caused by attackers would have to be compensated through investments. Investments incorporate the quantification of costs and benefits, where the costs of an investment simply include prices of required hardware, software, labour, etc., whereas benefits are more difficult to quantify. However, it is important that the value of the protected asset remains lower than the investments made to protect it.

Gordon and Loeb (2002) propose an economic model which uses marginal costs and marginal benefits to determine the optimal amount to invest in information security. They conclude that a company should only invest up until the marginal benefits of the investment are equal to the marginal costs. If the marginal benefits exceed the marginal costs, investments should be increased. In addition to this, the investments made to protect the security of a computer-based information system should protect the confidentiality, integrity and availability of the system, [79]. Therefore, in case of attack, the level of investments made should be separated into the 3 attributes, respective to an organisation's preferences to each.

An initial attempt to the development of such a model has been proposed by Ioannidis et al., (2009), in which the trade-off between the levels of confidentiality and availability of different organisations are considered in the presence of a shock on confidentiality. Integrity is neglected as corruption of data is assumed not to be a major issue in most cases. However, this does limit the validity of the model introduced.

We develop an improved model of Ioannidis et al. (2009) in Section 4 with the addition of integrity and shocks to each of the attributes and investment separately. Due to the lack of data, a systematic approach is taken to ensure the validity of the model. This then applied to different organisations, as part of a series of thought experiments, where we systematically explore the different dimensions of the model through mathematical simulations. This allows for the further prove of the correctness and efficiency of the model. Details of these are given in Section 4.

Optimal patching frequencies have been another popular area of research in the security economics literature, such as [46, 16, 191], amongst others. Ioannidis et al. (2012), for example, also produced some work, based on the previous paper, to derive the optimal patching frequencies for (ir)regular patches in military and financial organizations. By using utility theory and taking the patch arrivals to be shocks to the confidentiality and availability of the system, they find the optimal patching frequencies through optimization. They find that out-of-cycle patching is cost sensitive and its deployment is dependent upon an organisation's preferences. In the case of client patching, which faces frequent and low impact threats, it would not be optimal to patch on arrival. However, for network patching, which encounters high impact but low frequency threats, military organisations would tend to patch on arrival whereas financial organisations exhibit a high degree of sensitivity to it.

Optimal timings for investments have also been addressed in Williams et al. (2012), where a utility theoretic approach is used to find a quadratic approximation to an analytical solution in the presence of existing and future threats. They suggest that organisations which value confidentiality and availability more than the level of investment will have more frequent investment cycles as compared to those who value the investments in information security more.

Game theoretical concepts have also been applied to such problems to obtain various results. Cavusoglu et al. (2008) use a game theoretic approach to model the strategic interaction between a software vendor and a firm using that software to find the optimal frequency of patch updates, whilst Augusta and Tunca (2006), study the effect of user incentives on software security in a network of individual users under costly patching and negative network security externalities.

Other game theoretic applications of security economics include Grossklags and Johnson (2009), who have studied the impact of individual security investment decisions on the entire internet population through different security game paradigms, such as a total effort security game, in which an individual's utility depends on the average protection level of the network, or a weakest-link security game, where the utility of a node depends on the minimum protection level among all individuals in the network. They describe that, in the latter case, an attacker is able to compromise an entire security network once it is able to get past the perimeter of the organisation, due to inconsistent security policies.

Gao et al. (2015) define a Stackelberg game and Nash Equilibrium where a security provider and firm are the players, each including a level of hacker attack in their mathematical representation. They find that, under the Stackelberg security game, the firm invests less, the security provider invests more, and the security breach probability rate is smaller than under the Nash equilibrium. They also find that the firm has a higher payoff under the Stackelberg security game than in the Nash Equilibrium.

Here, the hacker attacks have been linearly defined to diffuse over time in the form of a differential equation, which takes into account the rate of diffusion without security measures and the investment rates for both the firm and security provider. However, this representation may not always be realistic, hence the need for a more accurate representation of the attacker arises.

We find from the literature that there exist more models to understand the behaviour of users in a system in the economic settings than understanding the attackers' incentives of that system. We believe that an economic model, which provides an integration of these two concepts, would provide a more comprehensive insight and understanding of security and its associated defence strategies.

3 Distributed Systems

We begin by introducing the way we have chosen to represent our underlying network model, with its associated mathematical representations, and apply it to a number of different examples to give the reader a better understanding of its applicability to different systems. The versatility of this framework will allow us to represent the underlying network model for both the attacker and economic security models in Sections 7 and 6. That is, the components of this framework can be used to capture various elements in the models proposed in later sections.

The general structure of a systems security network usually consists of a set of vertices, connected through edges, which are protected by different security policies. Each of the vertices represent different locations in a system, and each of the edges show any connections that may exist between them. These could either be physical, such as corridors between two rooms, or virtual, such as wireless connections between two separate computers.

Each of the locations in a system would contain different assets that would need to be protected, such as sensitive data, keys or employees' valuables. Therefore, some degree of security would have to be incorporated within the system, such as access controlled security barriers, locks on doors, or requirement of passwords, to protect these assets from malicious attackers.

To model this network structure, we use the classical theory of distributed systems as presented by Coulouris et al. (2000). Here, 4 main components of a system are introduced: locations, resources, processes and environment. Definitions of each of these components are given in Section 3.1.

We can then use methods from process calculus, graph theory and probability theory to set up a mathematical model to represent each of these components in Section 3.2.

We then present 3 different real life examples and apply them to our distributed systems model in Sections 3.3 – 3.5 to help the reader better understand the ideas presented in our model. In each of the examples, we describe their general architecture and set up and then move on to show how they are applied to our distributed systems framework.

We start off our set of examples with describing distributed database management systems which already have a structure similar to our systems framework. We give a brief introduction on the different types of DDBMS available and apply one of these to our distributed systems framework. We then move on to our second example of SecureDrop, which consolidates the notion of environments and informally captures the idea of interfaces. Lastly, we give an example of an employee's journey from their home to the office, as in [42], to provide a more in-depth view of the concept of interfaces and environments in our distributed systems framework.

3.1 Definitions

We start this section with definitions of the main components of the distributed systems framework. The main components of a system are given by their locations, resources, processes and environments, whose definitions are as follows:

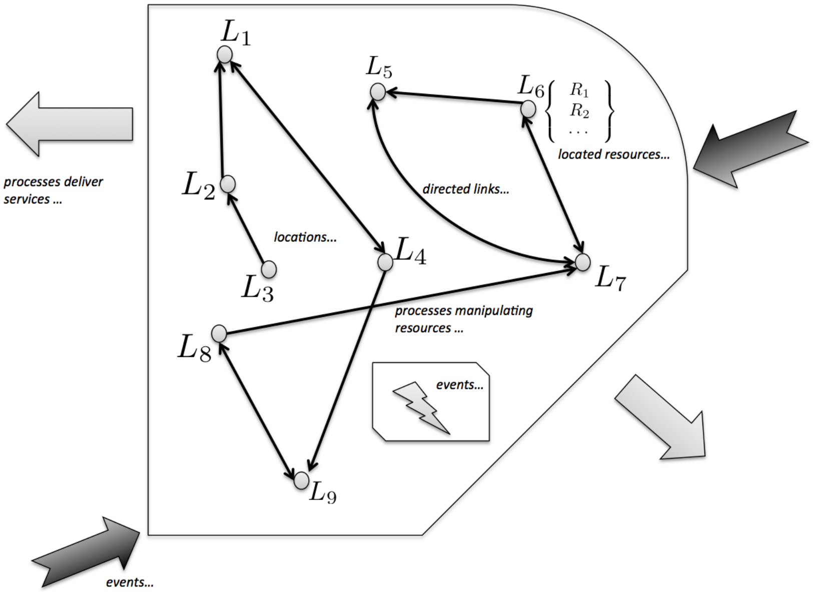


Figure 3.1: Generic example of a distributed systems model

Locations: Any physical or logical places in a network in which resources reside, such as locations in a computer memory, lockers, or rooms inside buildings

Resource: Components of a system which can be manipulated, e.g., consumed, read, modified, destroyed. These could be ID badges, sensitive documents, memory, money etc.

Processes: These model the dynamic parts of the system by manipulating resources to deliver services. For example, scanning of bags at X-Ray machines at airports, opening of doors as a result of a correct show of ID, etc.

Environment: The place in which each of the mentioned aspects can be found, and from which services can be delivered and received from different systems. For example, if we take a single room in a building to be our system, its environment could be the remaining rooms on the same floor (which would be systems themselves) or even the whole building.

Mathematical representations of these components using a combination of graph-like structures, process calculus, bunched logic and probability distributions are given in the following sections. A generic example of a system model (similar to [53]) is provided in Figure 3.1 to aid the reader in his understanding of each of the components.

3.2 Mathematical Representations of Components

Before we dive into the explanations of the different mathematical representations of the components, it is important to set out a few of the basic combinators in process calculus using Milner's synchronous calculus of communicating systems, SCCS [127].

We begin by defining some basic combinators in SCCS. Let $\mathtt{Act}$ be a monoid which contains actions a, b, where $ab \in Act$ is an action and there exists a unit 1 such that, for any $a \in \mathtt{Act}$, $1a = a = a1$. Let E, F be processes which are built up from actions using some combinators, we can then define the action prefix, concurrent composition and non-determinist choice as in Table 3.1.

We use the notion of bisimulation to describe the notion of equality. We say that

Type	Mathematical Example	Meaning
Action Prefix	$\frac{}{a{:}E \xrightarrow{a} E}$	An action a occurs and the process then evolves as E
Concurrent Composition	$\frac{E \xrightarrow{a} E' \quad F \xrightarrow{b} F'}{E \times F \xrightarrow{ab} E' \times F'}$	Two processes, E and F evolve parallel to each other with actions a and b.
Non-deterministic Choice	$\frac{E_i \xrightarrow{a_i} E'}{E_1 + E_2 \xrightarrow{a_i} E'}$, $i = 1,2$	Choosing to evolve one of two disjunctive components of a process with action a_i

Table 3.1: Basic combinators in Milner's synchronous Calculus of Communicating Systems (SSCS).

two processes, E and F are bisimilar, written as $E \sim F$, if

1. for every action, a, such that $E \xrightarrow{a} E'$, there is a F' such that $F \xrightarrow{a} F'$ and $E' \sim F'$ and,
2. for every action a such that $F \xrightarrow{a} F'$, there is an E' such that $E \xrightarrow{a} E'$, and $E' \sim F'$

In other words, each process imitates the behaviour of the other.

We can now begin to describe each of the components of the distributed systems model in the following sections.

3.2.1 Resources and Processes

We begin by using the resource semantics of O'Hearn and Pym's bunched logic, [53]. The general idea consists of two resource elements that can be combined and compared together to form a new element or to determine which is greater. This basic set-up has proven to be very useful when formulated mathematically, [15, 51, 42].

We assume the following basic properties of a resource as in [51, 52, 53]:

- A basic collection of resource elements, including a zero element,
- A notion of combination of resource elements,
- A notion of comparison of resource elements.

These properties can be formally defined as preordered, partial commutative monoids of resources, $(\mathbf{R}, \circ, \mathbf{e}, \leqslant)$, where $\mathbf{R}$ is a carrier set of resource elements, $\circ$ is a partial monoid composition, with unit $\mathbf{e}$, and $\leqslant$ is a preorder on $\mathbf{R}$.

Now that we have defined the notation for resources, we can start to describe the notion of processes.

As previously defined, processes model the dynamic parts of the system by manipulating the resources to deliver services. In order to represent this interaction, we first define the notion of processes using process algebra, [126, 91, 90, 73], which will be modified to take into account resources (and later, locations) to allow them to co-evolve.

We begin by letting `Act` be a monoid of actions with composition ab of elements a and b with unit 1 and we let E be a process with process variable X, where $a \in \texttt{Act}$. We can then define the grammar for process E as:

$$E ::= X|a|a : E|\sum_{i\in I} E_i|E \times E|\text{fix}_i X.E|(\upsilon R)E$$

where X is a process variable, a is an action, $a : E$ is the action prefix, $\sum_{i\in I} E_i$ represents the sum of i processes, where I is is any set. $E \times E$ is the concurrent product, $\text{fix}_i X.E$ takes X and E to be tuples and the ith component of the tuple is taken, and $(\upsilon R)E$ is the hiding operator which allows for the integration of the resource and processes.

We can now take resources, R, to co-evolve with process, E, with action a to represent the interaction between them. We describe this co-evolution as

$$R, E \xrightarrow{a} R', E' \tag{3.1}$$

conforming to the partial modification function, $\mu : (a, R) \mapsto R'$, which determines how an action a evolves the resource R to R' and process E to E'. This function is required to satisfy the following conditions:

- $\mu(1, R) = R$, where 1 is the unit function
- if $R \circ S$ and $\mu(a, R) \circ \mu(b, S)$ are defined, then $\mu(ab, R \circ S) = \mu(a, R) \circ \mu(b, S)$.

The meaning of these combinators is given by structural operational semantics (SOS) [3, 141], which has found considerable application in the theory of concurrent processes, [125, 19, 1]. Some operational semantics are given by the rules in Table 3.2.

The previously mentioned hiding operator can now be given as

$$\frac{R \circ S, E \xrightarrow{a} R' \circ S', E'}{R, \nu S.E \xrightarrow{\nu Sa} R', \nu S'.E'}$$

Type of Operational Semantics	**Mathematical Representation of Operational Semantics**
Action Prefix	$\dfrac{}{R,a{:}E \xrightarrow{a} \mu(a,R),E}$
Concurrent Composition[1]	$\dfrac{R,E \xrightarrow{a} R'E' \quad S,F \xrightarrow{b} S',F'}{R \circ S, E \times F \xrightarrow{ab} R' \circ S', E' \times F'}$
Sum	$\dfrac{R,E_i \xrightarrow{a} R'E'}{R,\sum_{i \in I} E_i \xrightarrow{a} R',E'}$
Recursion	$\dfrac{R,E_i[E/X] \xrightarrow{a} R',E'}{R, fix_i X.E \xrightarrow{a} R',E'}$

Table 3.2: Table showing different structural operational systems.

as in [51, 54]. That is, the resource S becomes bound to the process E.

As mentioned in the introduction, the notion of resources will not explicitly be expressed in our final model. Rather, we will model the protection of the properties of the resources in terms of their levels of confidentiality, integrity and availability and the investments involved.This will be explained in more detail in Section 4.

3.2.2 Location

We begin our treatment of location by outlining some of its basic requirements. These include a collection of atomic locations, which are the basic places in a system that generate a structure of locations, and the notion of (directed) connections between locations, which describe the topology of the system. We can also go on to consider sublocations (which respects connections), and a notion of substitution (of a location for a sublocation) that respects connections. This will provide a basis for abstraction and refinement in our system models, [53, 42]. Note that the treatment of locations in this way seems not to lead to a process calculus with operational behaviour that is more expressive in absolute terms. However, it does allow for more logical expressiveness and allows for the simplification of the construction of a wide range of systems.

The resulting calculus provides us with a transition system given by a judgement of the form $L, R, E \xrightarrow{a} L', R', E'$, where a is an action, L, L' are location environments, $R, R'$are resource environments and E, E' are processes used to control the evolution. Again, we use a modification function, μ, to determine the evolved location L' and resource R' with action a as $\mu(a, L, R) \mapsto (L', R')$. The action prefix can then be defined as

$$\frac{}{L, R, E \xrightarrow{a} L', R', E'} \quad \text{where} \quad L', R' = \mu(a, L, R)$$

Details of some of the structural operational semantics with locations can be found in Table 3.3.

Type of Operational Semantic	**Mathematical Representation of Operational Semantics**
Action Prefix	$\frac{}{L,R,a{:}E \xrightarrow{a} \mu(a,L,R),E}$
Sum	$\frac{L,R,E_i \xrightarrow{a} L',R'E'}{L,R,\sum_{i\in I} E_i \xrightarrow{a} L,R',E'}$
Fix	$\frac{L,R,E_i[E/X] \xrightarrow{a} L',R',E'}{L,R,fix_i X.E \xrightarrow{a} L',R',E'}$
Frame	$\frac{L,R,E \xrightarrow{a} L',R',E'}{L,R\circ S,E \xrightarrow{a} L',R'\circ S',E'}$

Table 3.3: Structural Operational semantics

We represent the notion of locations as a directed graph in our model, where each of the vertices represent different locations, which contain resources, and each of the directed arrows show the links between two locations. These could either be virtual links, such as wireless networks connecting one computer to another, or physical links such as corridors between two rooms or pathways from one location to another. The use of directed links will be useful when considering one-way connections between locations, such as receiving (spam) e-mails, or only being able to enter buildings from one entrance and exit from another.

More detailed of the representation of locations in the model will be given in Section 4.

3.2.3 Environment

Now that we have described the mathematical representations of the resources, processes and locations of our model, we can move on to define the notion of environment.

We take the environment to encapsulate a system with locations containing resources that get manipulated by processes. The systems we model do not exist in isolation, rather, they interact with their environment to which services can be delivered and from which they can be received. This includes both the external factors which have an impact on the system of interest as well as the internal parts of the model that we do not need to model in detail. Rather, we represent these parts by capturing the incidence on the model of events from the environment stochastically. For example, a negative exponential distribution may be used to model the arrival of agents at the entrance that marks the outer boundary of a model [147].

The interaction between two separate environments will be modelled as a composition of two separate systems models which interact with each other at the location, process and resource levels. To enable this composition, we use the idea of interfaces, as in [42], which define the locations, and their appropriate actions, where models fit together.

We begin by representing a model using a location graph, $\mathscr{G}(\mathscr{V}[\mathscr{R}],\mathscr{E})$, with a set of vertices, $\mathscr{V}$, representing the locations of the model, and a set of directed edges, $\mathscr{E}$ giving the connections between the locations. Each of the vertices are labelled with resources $\mathscr{R}$ and each of the actions in a model are contained in a set $\mathscr{A}$.

A model also contains a set of located actions, Loc_a, where a located action $\ell \in Loc_a$ is given by an ordered pair $\ell = \{a \in \mathscr{A}, v \in \mathscr{V}\}$. These located actions are associated with probability distributions: $Env : Loc_a \to ProbDist$, which are brought into existence during the execution of the model by sampling them from these.

We can then move on to represent the interaction between two separate environments using the notion of interfaces as explained above.

We take an interface $Int \in \mathscr{I}$ on a model to be a tuple, (In, Out, h) of sets of input and output vertices, where $In, Out \subseteq \mathscr{V}$. The sets of input and output vertices in interfaces must be disjoint, that is:

$$\bigcap_{i \in \mathscr{I}} In_i \in In = \emptyset \quad \text{and} \quad \bigcup_{i \in \mathscr{I}} Out_i \in Out = \emptyset$$

Now that we have set up the abstract framework, we can define a more detailed notion of a model, by including interfaces.

Definition 1. *A model $M = (\mathscr{G}(\mathscr{V}[\mathscr{R}],\mathscr{E}), \mathscr{A}, \mathscr{P}, Loc_a, \mathscr{I})$ is a tuple that consists of a location graph $\mathscr{G}$, a set of actions $\mathscr{A}$, a set of processes $\mathscr{P}$, a set of located actions Loc_a, and a set of interfaces $\mathscr{I}$*

A composition of two of these types of models, M_1 and M_2, with specific interfaces $\mathscr{I}_{1,1}, ..., \mathscr{I}_{1,j}, ..., \mathscr{I}_{1,n} \in \mathscr{I}_1$ and $\mathscr{I}_{2,1}, ..., \mathscr{I}_{2,k}, ..., \mathscr{I}_{2,m} \in \mathscr{I}_2$ can then be implemented through a composition operator $M_{1_{\mathscr{I}_{1,j}}} |_{\mathscr{I}_{2,k}} M_2$. This is defined using operation, $\oplus$, on each of the components of the model. Details of the definition of this operator for the composition of vertices, edges, actions, processes, locations and interfaces can be found in [42], with proofs of the existence of the commutative and associative properties of a model $M_{1_{\mathscr{I}_1}}|_{\mathscr{I}_2}M_2$.

We can now describe the application of our distributed systems framework to different real life systems using three different examples: Distributed Database Management Systems, SecureDrop and a General Office Security Model.

3.3 Example 1: Distributed Database Management Systems

A good first example to use to describe the real life applicability of our framework are the distributed database management systems due to the way their architecture is naturally set up. In this section, we give an overview of different types of DDBMS available to give the reader an idea of its structure, before describing one of them in terms of our distributed systems framework.

As explained by Ozsu (1991) and Bell (1992), a distributed database system is a collection of several logically related databases which are physically distributed in different computers (otherwise called sites) over a computer network. All sites in the distributed database have full control over themselves in terms of managing their data, and can inter-operate whenever required. The user of a distributed database has the impression that the whole database is local except for the possible communication delays between the sites. This is because a distributed database is a logical union of all the sites and the distribution is hidden from the user, ([145], [5] [59]).

Each day, thousands of transactions are carried out through the use of databases in many organisations' IT systems. There are various database management systems (DBMS) available, which include both packaged and open-source database suites, such as Oracle, IBM and Microsoft.

Each database can be linked to another to form one large distributed database environment, which can be broadly classified into homogeneous and heterogeneous distributed database environments as shown in Figure 3.2.

In homogeneous distributed databases, all the sites use identical DBMS and operating systems, which means that they are easier to design and manage. This approach allows for increased performance as each site is aware of all other sites and cooperates with other sites to process user requests. Types of homogeneous distributed databases include autonomous and non-autonomous, where each database is either independent, functions on its own and uses message passing to share data updates or data is distributed across the homogeneous nodes and a central master DBMS co-ordinates data updates across the sites, respectively.

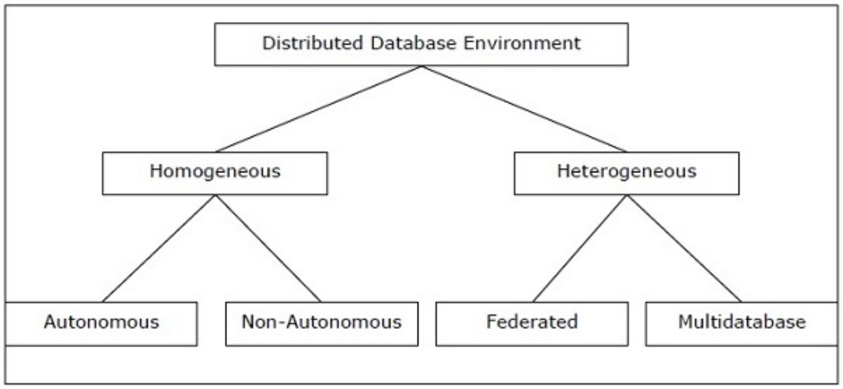


Figure 3.2: Classification of different types of distributed systems

Conversely, heterogeneous distributed databases have different operating systems, DBMS products and data models at different sites. This means that any query or transaction processes will be complex due to their dissimilar schemas and software. There are two types of heterogeneous distributed databases; federated and multi-database. Federated heterogeneous databases are all independent and are integrated together to function as a single database system, whereas multi-database systems employ a central coordinating module through which the databases are accessed.

DDBMS architectures are generally developed depending on three parameters: distribution, autonomy and heterogeneity.

Distribution refers to the physical distribution of data across the different sites, whereas autonomy describes the distribution of control of the database system and the degree to which each DBMS can operate independently. The degree of uniformity of data models, system components and databases are described by the heterogeneity of the architecture.

Some of the most common types of DDBMS architectures include Client-Server, Peer-to-Peer and Multi-DBMS, where each of these are further divided into different levels. To avoid repetition of the same ideas, we will only concentrate on the Client-Server architecture in the next section and represent one of its types in the distributed systems framework.

3.3.1 Client-Server Architecture

This refers to a two-level architecture where the functionality is divided into servers and clients. Server functions are mainly concerned with data management, optimization, query processing, etc, whereas client functions mainly include user interface as well as consistency checking and transaction management. There are two different types of client-server architecture: Single Server-Multiple Client and Multiple Server-Multiple Client. As the names suggest, the former describes an architecture where a single server is connected to a number of different clients and the latter indicates an architecture where multiple servers are connected to multiple clients. In both of these architectures a specific number of clients and servers are connected to each other. The client requests for a service and the server responds by providing them with the service, possibly obtaining it from a database it is connected to. The main focus of this architecture is to share information, where the data is stored in a cetralized server. One of the main problems with these types of architectures, however, is the possibility of getting bottlenecked, which would affect the efficiency of the system, [168, 67].

We will use the Multiple Server-Multiple Client architecture, as illustrated in Figure 3.3, to show its application in our distributed systems framework.

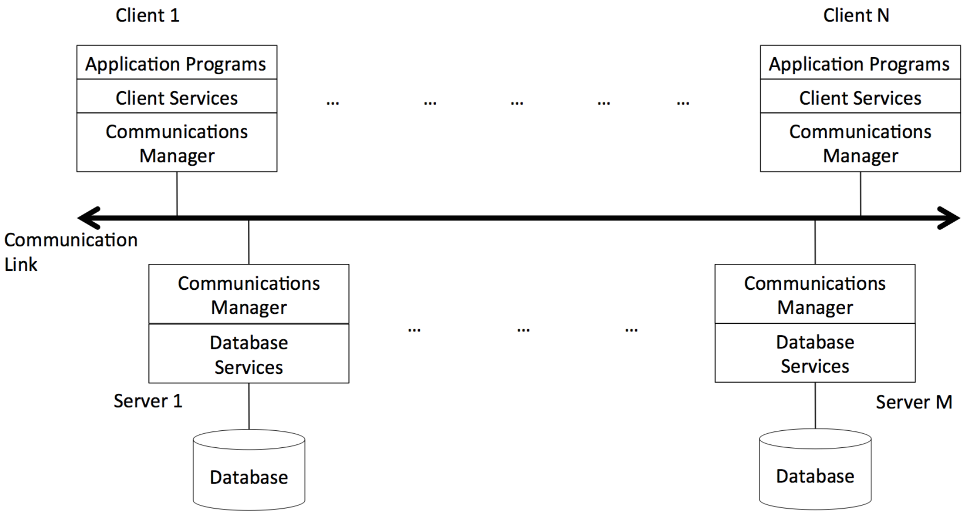


Figure 3.3: Multiple Server-Multiple Client architecture as in Tutorialspoint (2016)

3.3.2 Multiple Server-Multiple Client Architecture Representation in the Distributed Systems Framework

A representation of the Multiple Server-Multiple Client architecture in the distributed systems framework is illustrated in Figure 3.4.

Here, we have take each of the clients, servers and databases to be different locations in the system.

Each of these locations contain different resources, such as application programs, client services or database services.

We represent the communication links between the clients and servers and databases as bi-directed edges where requests can be sent from the client to the server and services sent back from servers to the appropriate clients. Each of the requests are taken to be resources which are then manipulated through processes to provide services. Note that the processes have now become implicit in the system's evolution over time.

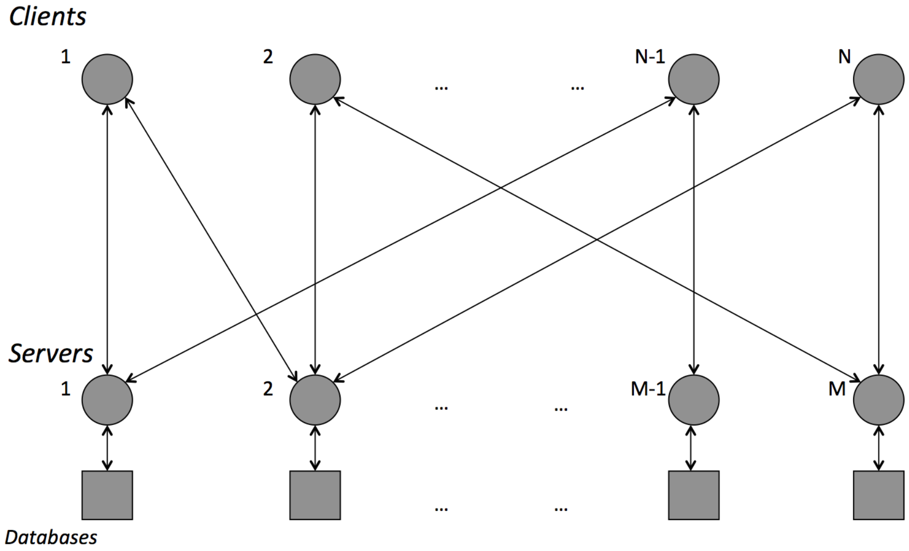


Figure 3.4: Multiple Server-Multiple Client architecture in the distributed systems framework, with some possible communication links between clients and servers depicted as directed edges.

3.4 Example 2: SecureDrop

Our second example is SecureDrop which allows us to demonstrate the idea of interfaces and environments more concisely, to aid the reader in his understanding of these concepts.

SecureDrop is an open-source whistleblower submission system that media organisations can use to communicate with anonymous sources who can provide them with useful documentations. It was originally developed by late Aaron Swartz and was taken over by Freedom of the Press Foundation in 2013.

This application environment consists of four main stations; Secure Viewing Station, Application Server, Monitor Server and the Journalist Workstation.

The Secure Viewing Station is a laptop running the Tails operating system from a USB stick that journalists use to decrypt and view the received documents whilst preserving their privacy and anonymity. This laptop has no direct connections to the internet or any other computers which may be connected to the internet. The Application Server uses the Ubuntu server to run two segmented Tor hidden services; the source and journalist interface. The source connects to the source interface to send messages and documents to the journalists whilst the journalist connects to the journalist interface to download the encrypted documents safely and respond to the source. The Monitor Service also uses the Ubuntu server to monitor the Application Server using OSSEC, which actively monitors all activities on the system.

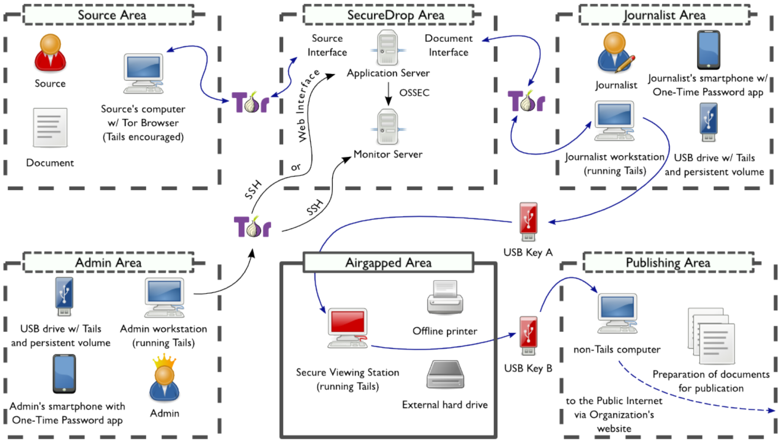
SecureDrop consists of four main components: the server, the administrators, the sources and the journalists. It runs on two main servers; the Application and Monitor Server, where the former runs the core SecureDrops software and the latter keeps track of the the application server and sends out alerts in case of problems. These servers are usually located physically inside the newsroom and are connected to firewall appliances.

The administrators manage the SecureDrop servers. They use Admin Workstations using Tails and and connect tot the Application and Monitor servers over authenticated Tor Hidden Services.

A source is a person who submits documents and messages through the Tor browser to access the Source Interface. Each of the submissions are encrypted on the Application Server as they are uploaded.

Finally, journalists working in the newsroom use two machines to interact with SecureDrop. These include the journalist workstation which connects to the journalist interface using Tails. They download the encrypted submissions and copy them to a transfer device, such as a USB, and access those by connecting their device on the air gapped secure viewing stations which contain the key to decrypt them. Journalists can then choose to read, print or otherwise prepare the documents for publication.

An illustration of this process is given in Figure 3.5.



Source Area
Source
Source's computer w/ Tor Browser (Tails encouraged)
Document
SecureDrop Area
Source Interface
Document Interface
Application Server
OSSEC
Monitor Server
Web Interface
or
SSH
SSH
Journalist Area
Journalist
Journalist's smartphone w/ One-Time Password app
Journalist workstation (running Tails)
USB drive w/ Tails and persistent volume
USB Key A
Admin Area
USB drive w/ Tails and persistent volume
Admin workstation (running Tails)
Admin's smartphone with One-Time Password app
Admin
Airgapped Area
Offline printer
Secure Viewing Station (running Tails)
External hard drive
USB Key B
Publishing Area
non-Tails computer
Preparation of documents for publication
to the Public Internet via Organization's website

Figure 3.5: Overview of the SecureDrop Infrastructure showing the connections between Source, SecureDrop, Journalist, Admin, Airgapped and Publishing Area.

3.4.1 SecureDrop Representation in the Distributed Systems Framework

We can represent the SecureDrop system in the distributed systems framework as 6 different environments containing locations and resources, as illustrated in Figure 3.6.

We let the documents submitted by the source/received by journalist be the resources. These are manipulated through processes, such as Tor, to keep them secure. Note that, again, the processes have not explicitly been shown in the figure as these are now embedded in our model.

In addition to the servers, workstations and devices being represented as locations, we also let USB Key A and USB Key B be locations, as they contain the encrypted/decrypted documents (resources).

The possible routes the resources can take are tracked through the directed edges between locations, starting off from Location 1 and ending in Location 14.

Each of the environments in this system are composed through interfaces which are represented through the interaction between the connecting locations and the changes that occur in the resources (encrypted/decrypted, moved, etc.).

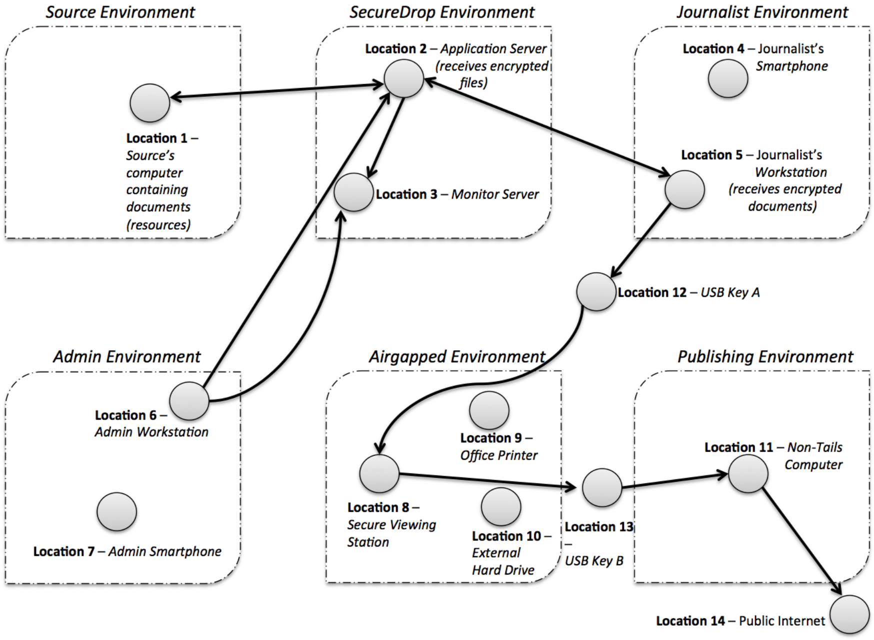

Source Environment
SecureDrop Environment
Journalist Environment
Location 1 – Source's computer containing documents (resources)
Location 2 – Application Server (receives encrypted files)
Location 3 – Monitor Server
Location 4 – Journalist's Smartphone
Location 5 – Journalist's Workstation (receives encrypted documents)
Location 12 – USB Key A
Admin Environment
Airgapped Environment
Publishing Environment
Location 6 – Admin Workstation
Location 7 – Admin Smartphone
Location 9 – Office Printer
Location 8 – Secure Viewing Station
Location 10 – External Hard Drive
Location 13 – USB Key B
Location 11 – Non-Tails Computer
Location 14 – Public Internet

Figure 3.6: The SecureDrop Infrastructure in the Distributed Systems Framework.

3.5 Example 3: General Office Security Model

Our final example encompasses a more detailed illustration of each of the components in the distributed systems framework. We use a general office security model as introduced in Caulfield and Pym (2015), which displays a typical journey for employees working in an office.

It captures the journey of employees, who work in an office, from their homes, via a transport system, to their office building. Upon arrival, they would be required to get through some type of access controlled entrance to get inside the office building. This could be access controlled barriers which provide access to the employee if a correct ID card is shown, or full body scanners with accompanying security guards to carry out a full body check before entry. Once the employee enters the building, they would continue with their daily activities within the building.

This journey is naturally divided into three models. The first model contains details of the journey outside the office building, for example. from the employee's home to the office. The second is the entrance of the office, and the third is taken to be the back office where the employee carries out its daily duties.

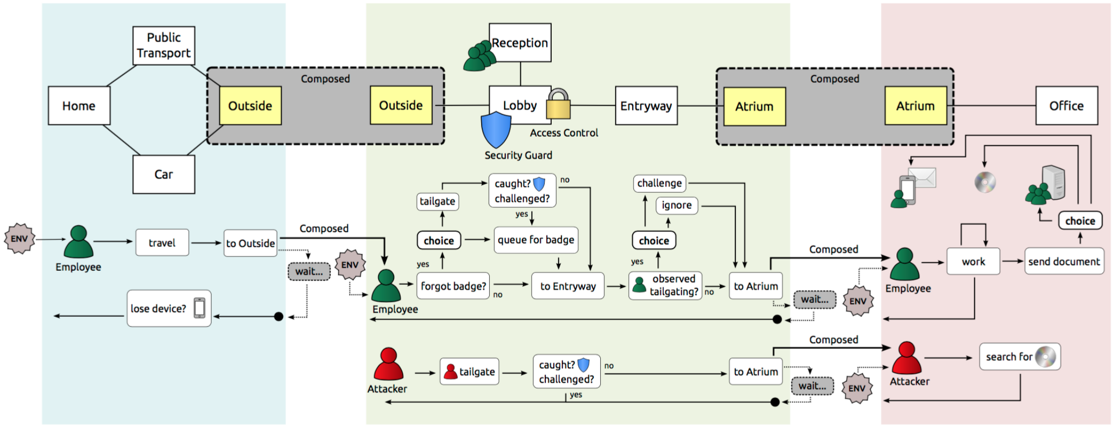


Figure 3.7: Example of an office model network from Caulfield and Pym (2015)

Caulfield and Pym analyse three different security aspects an organisation could face. These are separated into three different models where the first model looks at tailgating behaviour of employees and attackers at the entrance of an office building. The second model depicts the behaviour of employees inside the office when making decisions about how to share confidential documents with other employees and the third model looks at the loss of devices outside the office which could contain confidential data. These models can then be composed to examine the interactions between parts of the organization's security policy using interfaces, which specify the locations that are shared between the models, such as the Outside and Atrium as in Figure 3.7.

3.5.1 General Office Security Model Representation in the Distributed Systems Framework

We represent this model in the distributed systems framework with 3 different environments. Environment 1 encompasses all the events that take place outside the office building, such as the employee's home or travel arrangements. Environment 2 models the different places an employee can go at the entrance of the building, such as the lobby or reception and Environment 3 models the events taking place inside the office.

Each of the environments are composed through shared locations: Environment 1 and 2 are composed through the 'outside' location and Environment 2 and 3 are

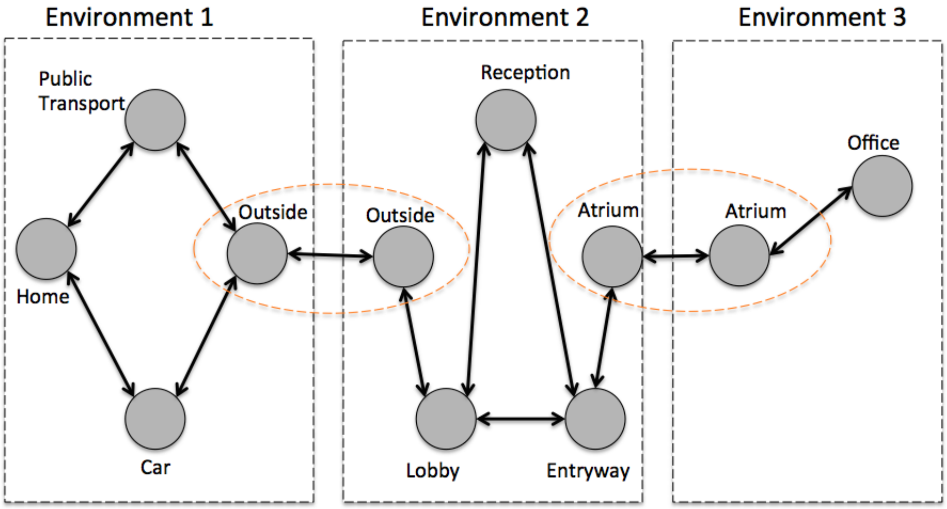


Figure 3.8: Example of an office model network from Caulfield and Pym (2015) in the Distributed Systems Framework.

composed through the ’Atrium’ location. These compositions are possible due to interfaces as explained previously, which are illustrated as dashed ovals in Figure 3.8.

Each of the vertices in this figure represent the different locations considered, such as ’Car’, ’Entryway’ or ’Office’, and each of the edges represent the possible paths that can be taken between the locations, such as a path from a type of public transport to the lobby or from the entryway to the employee’s office. Each of these locations contain different resources to be protected, such as sensitive data on computers or employees’ valuables.

Once again, the processes have become implicit in our modelling framework, such as the opening of access controlled barriers or movements from home to lobby.

4 Initial Economic Security Model

In this section, we introduce an economic security model, inspired by [94], that will allow us to gain a better understanding of the trade-offs between the security attributes and the amount of investment required to keep an organisation secure.

We use loss functions and impulse-response functions to understand how an organisation, that behaves under a specific set of assumptions, would return back to its equilibrium state once it has been attacked. The model explores this behaviour under a dynamic, stochastic shock whose magnitudes are taken to be random for now. We enrich this model in the next section by integrating it with the distributed systems model to have an underlying framework which describes the organisation's security network. This enrichment will allow us to observe the possible movements of attackers in the system and the damages these could cause at each location by using the game theoretic model developed in Section 7.7.

In what follows, we explain the details of the set up of the model and the trade-offs between the security attributes under the given assumptions.

As mentioned previously, organisations must maintain certain levels of confidentiality, integrity and availability to ensure the overall functionality of their system. We consider system confidentiality to mean the prevention of sensitive information reaching malicious users, while ensuring that the right people are still able to access it. Popular measures enhancing confidentiality include two-factor authentications (user ID and passwords), locks on doors, or enforcement of file permissions so that only authorised agents can access the file.

Maintaining the integrity of an organisation involves the safekeeping of data accuracy. It requires the prevention of any data alteration initiated by unauthorised users. To maintain integrity, systems' managers require data encryption or checksum, a protocol that compares the number of bits sent in a transmission to bits arrived.

Finally, system availability refers to the system's property that enables reliable and predictable access to information by authorised agents. This could be provided by regular system upgrades (to prevent it from crashing) and/or creating back up copies of data.

Each organisation will have different requirements for each of the security attributes. For example, a private company might invest heavily in access-controlled barriers at the entrance of their building to only allow entry to those with the correct ID badge, whilst a bookstore might direct security investment in CCTV cameras to catch thieves.

We take C, I and A to be the levels of confidentiality, integrity and availability of the system and $\dot{C}$, $\dot{I}$ and $\dot{A}$ to be their rates of change over time, respectively. We represent the existing level of security investments as K and the rate of change of investment as $\dot{K}$.

In the following subsections, we introduce relationships between the trade-offs of each of the security attributes and investment, as an extension of the model presented in [94], under a given set of assumption. We will then move on to describing loss functions and their application to our model, as well as impulse-response functions. We then define the stability conditions for our model and give details of the sensitivity analysis carried out to validate our model. Finally, in this section, we apply our model to 4 different organisations as part of thought experiments and discuss the results.

4.1 Complexity

We start by defining the complexity of the system, which here we simply take to be the degree of interconnectivity between locations in a system. We set the equation for the complexity in the system to be as [94]:

$$R = \frac{1}{1-\xi}, \quad \text{for} \quad \xi \in [0,1) \tag{4.1}$$

where ξ represents the level of interconnectivity of the system; that is, if $\xi = 0$, there is no interconnectivity within the system, therefore the system complexity is trivial. However, as the interconnectivity of the system tends to (i.e. it becomes more interconnected) the complexity of the system tends to infinity.

This model of complexity simply gives and indication of the scale of the attack surface. We incorporate a richer representation of the structure of the system, in terms of the distributed systems framework, in Section 6.

4.2 Investment

We define the rate of change of security investments in the system as follows:

$$\dot{K} = \nu\dot{R} - (\chi\dot{C} + \lambda\dot{I} + \eta\dot{A}) \tag{4.2}$$

where $\{\nu, \chi, \lambda, \eta\}$ measure the impact of each of the factors on the additional expenditure on information security. This relationship can be explained as follows:

1. Since we have taken the complexity, R, of a system to be a measure of the interconnectivity between locations in a system, we assume that it would imply that a positive change in the complexity of the system represents an increase in the number of locations, and in turn, the number of resources, within an organisation. These could be in the form of additions of new offices in a building or new computers or laptops provided by the company. This would suggest a larger 'attack surface', therefore more controls would be needed to protect the 'new' locations within the organisation, such as firewalls, access controlled barriers or new locks on doors. These controls would necessitate extra expenditure for the security managers to ensure the security of the organisation, causing a positive change in the investments also. Therefore, we assume a positive relationship between the change of security investments made in the organisation and its complexity.

2. The presence of technological advances, that reduce vulnerabilities, would suggest that no additional investments in C, I, and A would be required, therefore forming a negative relationship with $\dot{K}$. For example, the recent research report by Neustar on DDoS attacks, shows that targeted organisations experienced a minimum revenue risk disruption in excess of \$2.2 billion dollars in 2016, which is more than \$2.5 million averaged across 849 organisations, [133]. This indicates the necessity of investing a significant amount in order to regain control and protect the organisation from similar attacks in the future.

   Another study, as mentioned in the European Parliament's 'Data and Security Breaches and Cyber-Security Strategies in the EU and its International Counterparts' report, which focused on the outcomes of breaches of personally identifiable information in 117 US firms between 2005 - 2010, found that the average cost per data breach was approximately \$2.4 million, [44]. This indicates the requirement of investments to be made in case of data breach in these countries.

   Therefore, as the level of protection of the security attributes decreased, increased investments in various security measures were made to re-secure the organisation's security. Hence, we assume a negative relationship between these.

4.3 Confidentiality

We define the time evolution for the protection of the confidentiality of the system as:

$$C = -\alpha \int_{t_0}^{t} \dot{A} dt + \beta \int_{t_0}^{t} \dot{K} dt + \tau \int_{t_0}^{t} \dot{I} dt - \omega \int_{t_0}^{t} \dot{R} dt + C_0 \tag{4.3}$$

where $\{\alpha, \beta, \tau, \omega\} \in [0, 1]$ depict the impact of each of the factors on the confidentiality of the resource at location i for $t_0 < t$.

The reasoning behind each of the relationships, along with the assumptions made, are explained below:

1. **Confidentiality vs. Availability**:

   - An increase in the protection of the availability, $\int_{t_0}^{t} \dot{A} dt$, of the system would suggest that there has been an increase in the number of resources/services available. We assume that this would make it more difficult to preserve the protection of the confidentiality of these as there now are many different resources/services to be protected, with differing types of controls. We base this assumption on the following events that have occurred in the past, as outlined in the following bullet points.
   - The cyber attack against the retail company Target in 2013 was mainly caused due to network access given to a third-party vendor (increased availability) which did not appear to follow broadly accepted information security practices. This allowed the attackers to infiltrate Target's network and steal financial and personal information (decreased confidentiality) of over 100 million Target customers, [167].
   - The whistleblower attack by Katherine Gun, a former GCHQ translator, who leaked details of an operation to bug United Nations offices before the 2003 invasion of Iraq, [105, 106] is another reason for making this assumption. This was published on the front page of the Observer newspaper and cost Gun her job. The increase in the availability of information to the public lead to a drop in its level of confidentiality, conforming to the initial assumption of the relationship between the attributes.

     Therefore, we assume that as the availability of the resources/services increases, it will have a negative effect on the level of their confidentiality.

2. **Confidentiality vs. Investment**:

   - We assume that an increase in the security investments made, $\int_{t_0}^{t} \dot{K} dt$, would suggest a better protection of the confidentiality of the system, therefore allowing for the level of protection of the confidentiality of the resource to be increased.

  - For example, as more investments are being made to protect the confidentiality of patients' data in a medical research organisation, through either two-factor authorisation or encryption, it would become harder for attackers to gain access to them, therefore preserving the confidentiality of their data.
  - Additionally, a bank's investment in additional security measures, such as Hardware Security Models (HSMs), to ensure the confidentiality of their customer's information would cause a positive change in the investments as well as an increase in the level of confidentiality of customer's data. Therefore, we assume that there is a positive relationship between the change in security investments and the level of protection of the confidentiality of the system.

3. **Confidentiality vs. Integrity**:

    - We assume that confidentiality and integrity are aligned to some degree, which would allow us to assume a positive relationship between them. Increased integrity would imply that it is harder for unauthorised actors to make changes to a system including, for examThe use of Hardware Security Models (HSMs) in banks is an example of this. This well-established, highly secure method of authenticating payments uses digital certificates, which use Public Key Infrastructure (PKI), to enable the secure exchange of information. It allows trusted organisations, such as banks, to issue digital certificates to individuals or other organisations, [6]. Here, the confidentiality of the information to be exchanged, such as automated payments, is preserved due to the maintenance of the integrity (i.e. requirement of digital certificates to ensure that the information is coming from the right source).

4. **Confidentiality vs. Complexity**

    - We assume that a decrease in the complexity, $\int_{t_0}^{t} \dot{R} dt$, of the system would make it easier to maintain its confidentiality as there would be less interconnectivity between the different locations, therefore making it harder for the attacker to move between locations and damage other resources.

- This relationship is also established in Alpcan and Başar (2011) to set up their vulnerability model, where they assume that fewer connections between computers would suggest that it would be harder for attackers to gain access to them, therefore making them less vulnerable to attack.

  For example, if a virus is able to successfully gain access to one of the computers, it would be hard for it to directly spread itself to the others since they are not functionally connected via a network. This set-up would allow the organisation to continue maintaining the confidentiality of the other computers.

  Therefore, we assume that low levels of connectivity tend to be associated with higher levels of confidentiality, where ω is a measure of the impact of the complexity on the confidentiality of the system.

5. Once the system shuts down, we assume that a certain level of protection of the confidentiality of the system, C_0, is still maintained. For example, the disconnection of a laptop or computer from the internet would reduce its likelihood of getting attacked by a computer virus significantly, if not completely remove it. However, it would still be possible to be attacked physically by an intruder.

4.4 Integrity

The dynamics for the time evolution of integrity is given by

$$I = \sigma \int_{t_0}^{t} \dot{C} dt - \theta \int_{t_0}^{t} \dot{R} dt + \iota \int_{t_0}^{t} \dot{K} dt + (w_{I_1}(\phi_1) - w_{I_2}(\phi_2)) \int_{t_0}^{t} \dot{A} dt + I_0 \tag{4.4}$$

where $\{\sigma, \theta, \iota, \phi_1, \phi_2\} \in [0,1]$ show the impact of their respective components on integrity, $\{w_{I_1}, w_{I_2}\}$ are the weights associated with ϕ_1 and ϕ_2, respectively for $t_0 < t$.

The following assumptions are made regarding each of the relationships between integrity and the remaining security attributes, investment and complexity:

1. **Integrity vs. Confidentiality**:

   - An increase in the confidentiality of the system, $\int_{t_0}^{t} \dot{C} dt$, would prevent unauthorized access to the data, therefore allowing to maintain, if not increase, the level of integrity of the data. This would, therefore, cause a positive relationship between the two attributes.

- This relationship is captured in the Biba Integrity model, [27, 180], where a set of access control rules are set to ensure data integrity. Here, a user is only allowed to 'read up' (i.e. they must not read data at a lower integrity level) or 'write down' (i.e. they must not write to data at a higher level of integrity), which ensures that authorised users are able to read documents written by those who are in a higher position than them, but are only allowed to edit documents which are at a lower position than them. This ensures that only those who are allowed to access the information are allowed to edit them, thus helping in preserving its integrity.
- For example, let us consider the military chain of command, where their ranking is as follows: General > Lieutenant > Colonel > Major > Sergeant > Private. In this case, a General would be allowed to write orders to a Colonel who could issue them to a Major. This ensures that the original orders are kept intact and the mission of the military is protected ("read up"). Conversely, a Private would not be allowed to issue orders to his Sergeant, who may never be able to issue orders to a Lieutenant, therefore also protecting the integrity of the mission ("write down").

2. **Integrity vs. Complexity**:

   - Similar to its relationship with confidentiality, we assume that a decrease in the complexity, $\int_{t_0}^{t} \dot{R} dt$, of the system would make it harder for an attacker to move between locations and modify data, therefore maintaining their integrity.
   - For example, if a sensitive document is stored on two different computers, one which is connected to the internet and one which is completely disconnected from it, and if a hacker is able to get access to the one through the internet and alter it, she would not be able to gain access to the disconnected computer through the internet and alter that one too. Therefore, we assume that high levels of integrity tend to be associated with lower levels of complexity.

3. **Integrity vs. Investment**

   - A positive change in the level of security investments would suggest an increase in the level of protection of the integrity of the system as investments would have been made to increase the security of the system. This view is supported by many, including the managing director of Accenture security, who said that "making wise investments in innovation can certainly help make a significant difference when cyber criminals strike" when talking about recent research carried out with regards to which security investments make a difference, [20]. This includes the investments made in security measures to protect the integrity of data in organisations. Additionally, this view is supported by a survey carried out by CSI director, Robert Richardson in 2008, who found that out of the 144 respondents, there was an average financial loss of $288,618 per respondent due to cyber crimes, such as viruses damaging data, financial fraud, insider abuse and many more, [146]

- Let us also consider, as a hypothetical example, an organisation which requires its employees to manually enter data into the system, such as accountants, or Help Desk staff at banks. As humans, they are bound to make mistakes, which would affect the integrity of the data. The company could invest in training for these employees to ensure they correctly enter data or apply data validation rules in their system to restrict the values the employees can enter. This would cause a positive relationship between $\int_{t_0}^{t} \dot{K} dt$ and the levels of integrity of the system.
- Another example could be investing in different types of checksum. In order to ensure the data you have previously saved is the same as the data you open, you could apply a checksum to your data. This would ensure that the file has stayed the same as it is. In order to ensure a hacker is unable to hack this checksum, you'd want to invest in the most advanced checksum. Hence, this investment would cause an increase in the level of protection of the integrity of the data.

4. **Integrity vs. Availability**

   - The relationship between integrity and availability is more intricate. Initial thoughts would suggest that an increase in the availability of a resource would decrease its integrity, as the resources would be more readily available and therefore easier to modify. This would suggest that in order to protect the integrity of the system, the level of availability of the resource should be limited, resulting in a negative relationship.
   - For example, the recently shut down darknet marketplace AlphaBay sold illegal tickets such as hotel, plane or Disneyland tickets, for low prices on the Tor network, [11]. This breach in the integrity of the tickets was overcome by shutting down the marketplace as part of a multinational law

enforcement operation, [182]. Thus, ensuring the integrity of the tickets sold by limiting (in this case completely stopping) their availability..

- Another example could be having only one copy of a sensitive document and keeping it in a safe place with high encryption, it would make it harder for an attacker to gain access to it and find out the information on the document as there is only one copy available which is strongly protected, therefore protecting its integrity by limiting its availability to unauthorized people.
- However, for some cases, increasing the availability of the resources would help maintain their integrity. For example, Certificate Transparency allows website users or domain owners to identify maliciously or mistakenly issued certificates or certificate authorities that may have gone bad, [47]. This ensures that the integrity of the digital certificates is maintained, through increased availability.
- Therefore, we split the availability factor in the integrity model into two parts, whose parameter values are dependent on the type of resource or organisation considered:

  $$w_{I_1}(\phi_1)\int_{t_0}^{t}\dot{A}dt' - w_{I_2}(\phi_2)\int_{t_0}^{t}\dot{A}dt'.$$

  where the weights w_{I_1} and w_{I_2} represent the importance of the availability and non-availability of the resource with respect to its protection, respectively, with $\sum w_{I_n} = 1$, and ϕ_1, ϕ_2 capture the impact of availability on the integrity.

5. Similar to C_0, we use I_0 to represent the level of integrity maintained once the system shuts down.

4.5 Availability

Finally, we define the time evolution for availability as:

$$A = \gamma\int_{t_0}^{t}\dot{R}dt + \delta\int_{t_0}^{t}\dot{K}dt - \epsilon\int_{t_0}^{t}|\dot{C}|dt - \psi\int_{t_0}^{t}\dot{I}dt \tag{4.5}$$

where $\{\gamma,\delta,\epsilon,\psi\} \in [0,1]$ show the impact of each of the factors on availability for $t_0 < t$. The assumptions made for each of the relationships between availability and the remaining security attributes, investment and complexity are given below:

1. **Availability vs. Complexity**:

    - We assume that the complexity, $\int_{t_0}^{t} \dot{R} dt$, of a system directly determines its availability. As the system becomes more complex, its attack surface becomes larger and more interconnected, hence making it easier to gain access to it through several routes.
    - An example of this would be the distribution of a number of copies of a code to a locked safe containing valuables such as money or jewellery. As the number of people that are aware of the code increases, the probability of the code reaching a malicious person and stealing the valuables when the opportunity rises also increases.

2. **Availability vs. Investment**:

    - An increase in the resources within a system, such as the number of servers in businesses, number of clothing items in shops or the number of vaccinations in a hospital, would require a positive change in investment, $\int_{t_0}^{t} \dot{K} dt'$. Such additional investment would support an increase in availability of the resources, resulting in a positive association.
    - The protection of the readily availability of these resources to the appropriate people, we assume, would be maintained through security investments. Again, this point is supported by several recent researches carried out by organisations, such as [20, 44, 133]

3. **Availability vs. Confidentiality**:

    - The relationship between availability and confidentiality, $\int_{t_0}^{t} |\dot{C}| dt$, is more delicate. As mentioned before, an increase in confidentiality would make it harder to easily access information in a timely manner, thus reducing availability. For example, if access to a file on a computer is obtained through undergoing a number of different checks, it becomes harder to access it, therefore making it less available and, in effect, reducing availability.
    - The recently proposed SOTE model by the Norwegian Defence Research Establishment, [88], supports this assumption. By defining a set of restriction on who can access the resources at all times (increased confidentiality), the availability of the resources becomes limited, therefore reducing its availability in general.

- On the other hand, an exogenous decrease in the confidentiality level of a system would require the security manager to limit its availability to avoid any further damage. For example, a security manager's first response to an attack through the internet could be to disconnect the system from the internet. This would reduce the availability of the network surface area to the attacker and prevent any further attackers from entering the network, which again causes a negative relationship between the two attributes. This was the case in the Sony Pictures hack attack in 2014, [181], where a hacker group, by the name of 'Guardians of Peace' (GOP), leaked confidential data from the film studio including information about executive salaries at the company and un-released Sony films. The company's first response was to shut down Sony's entire computer system, including the network, Internet and any customer-facing sites to stop any further damage. Employees were also instructed to immediately shut down any devices connected to the Wi-Fi and not to download anything on the company lot or engage with any e-mails [155].
- We will, therefore, assume a negative relationship between confidentiality and availability, even for negative values of confidentiality. This relationship is captured by taking the the modulus of $\int_{t_0}^{t} \dot{C} dt$ to ensure a negative relationship between the two attributes at all times.

4. **Availability vs. Integrity**

   - We assume that a decrease in the levels of integrity, $\int_{t_0}^{t} \dot{I} dt$, of the system would suggest that security measures put in place to protect its resources from getting altered have been relaxed, therefore making it more easily available to unauthorised users (increased availability).
   - This assumption is also confirmed by the Clark-Wilson Model, which only allows authorised users to access and modify data in order to maintain its integrity. Therefore, as the level of protection of the integrity of the system increases, its availability becomes limited to only those who are authorised to access it.

4.6 Dynamic, Stochastic Shocks and Targets

System managers are required to set appropriate targets for each of the security attributes to maintain the security levels in their system. Investments would be made in people (i.e., training staff), processes (i.e., using ID badges to access rooms) and technologies (i.e., access controlled entrances) to achieve those targets and ensure optimal security levels within the organizations. We will postulate the behaviour of each of the organizations over time in the presence of stochastic, dynamic shocks using impulse-response functions.

These shocks will be added to each of the security attributes, as defined by equations (4.2)-(4.5), in the following form:

$$S_{*_t} = \rho_* S_{*_{t-1}} + \mu_{*_t} \quad \text{where} \quad * \in C, I, A, \dot{K} \tag{4.6}$$

Here, $\rho_* \in [0,1]$, is a measure of the persistence of the shock on C, I, A, or $\dot{K}$, at time $t-1$. As $\rho_* \to 1$ the system takes longer to recover. μ_{*t} is an innovation whose value is taken to be random for this section. In future work, a more accurate value for this can be calculated, as recommended in Section 8.

The time evolutions of the security attributes in the presence of shocks can now be expressed as follows:

$$C = -\alpha \int_{t_0}^{t} \dot{A} dt + \beta \int_{t_0}^{t'} \dot{K} dt + \tau \int_{t_0}^{t'} \dot{I} dt' - \omega \int_{t_0}^{t'} \dot{R} dt' + C_0 - S_{C_t} \tag{4.7}$$

$$\begin{aligned} I = \sigma \int_{t_0}^{t'} \dot{C} dt' - \theta \int_{t_0}^{t'} \dot{R} dt' + \iota \int_{t_0}^{t'} \dot{K} dt' + \\ (w_{I_1}(\phi_1) - w_{I_2}(\phi_2)) \int_{t_0}^{t'} \dot{A} dt' + I_0 - S_{I_t} \end{aligned} \tag{4.8}$$

$$A = \gamma \int_{t_0}^{t'} \dot{R} dt' + \delta \int_{t_0}^{t'} \dot{K} dt' - \epsilon \int_{t_0}^{t'} |\dot{C}| dt' - \psi \int_{t_0}^{t} \dot{I} dt' - S_{A_t} \tag{4.9}$$

$$\dot{K} = \nu \dot{R} - (\chi \dot{C} + \lambda \dot{I} + \eta \dot{A}) - S_{K_t} \tag{4.10}$$

The system responds to the sum of the deviations in the security attributes and change in investment as follows:

$$R = x(C - \bar{C}) + y(I - \bar{I}) + z(A - \bar{A}) + l(\dot{K} - \bar{\dot{K}}) \tag{4.11}$$

where x, y, z, l are the control variables for each of the deviations, C, I, A and $\dot{K}$, respectively.

We can now use loss functions to describe the losses produced at each time in the next section. We begin by introducing the idea of some common loss functions that can be found in the literature. We outline the reasons for the (non-)applicability of each of them to our model and describe the set up of the Hinge Loss function which we use in our model in Section 4.7.1.

4.7 Types of Loss Functions

A crucial element in all optimising problems, such as in decision theory, financial investment or forecasting is a loss (or cost) function. These describe the losses incurred by an organisation as a result of their deviations from target at each time period.

Lee (2007) describes a loss function as follows:

Definition 2. *At time t, a loss (or cost) occurs when a forecast, $f_{t,h}$ of a variable, Y_{t+h}, for h periods ahead, is different from the actual value at time $t+h$. We will denote the loss function of the forecast error, $e_{t+h} = Y_{t+h} - f_{t,h}$ as $L(e_{t+h}, t)$. [115]*

Note that here, the target level is the forecast value, $f_{t,h}$ and Y_{t+h} is the actual value at that time. The value of the losses incurred is given by the difference between these terms.

Granger (1999) further discusses the following set of properties for a loss function:

1. $L(0) = 0$ (i.e., no error = no loss)

2. $L(e) > 0$ for $e \neq 0$

3. $L(e)$ is monotonically non-decreasing as e moves away from zero so that $L(e_1) \geqslant L(e_2)$ if $e_1 > e_2 > 0$ and if $e_1 < e_2 < 0$.

He shows that for a given loss functions, $L_1(e)$ and $L_2(e)$, further examples of loss functions can be generated as follows:

1. $L(e) = aL_1(e) + bL_2(e)$ for $a, b \geqslant 0$ will be a loss function,

2. $L(e) = L_1(e)^a L_2(e)^b$ for $a, b > 0$ will be a loss function,

3. $L(e) = 1(e > 0)L_1(e) + 1(e < 0)L_2(e)$ will be a loss function, and

4. If $h(\cdot)$ is a positive monotonic non-decreasing function with $h(0)$ finite, then $L(e) = h(L_1(e)) - h(0)$ will also be a loss function.

Since higher errors would tend to imply higher losses produced, convex loss functions are often used to represent this behaviour. Therefore, we will only analyse convex loss functions in this thesis. The reader interested in the different types of concave loss functions is directed to [89].

Mean Squared Error (MSE)

$$L(e_{t+h};\alpha) = \alpha e_{t+h}^2$$

where α is a parameter value.

This loss function has become the most popular in the literature due to its mathematical tractability, symmetric nature and differentiability. It is monotonically increasing and homogeneous. The latter property will be useful when solving for optimal forecasts, since the optimal forecast will be constant for different values of α.

However, a main problem with this function is its behaviour with any outliers in the data. These would be heavily punished by squaring the error. Therefore, any outliers would need to be filtered out first before using this form.

Mean Absolute Error (MAE)

$$L(e_{t+h};\alpha) = \alpha|e_{t+h}|, \quad \alpha > 0$$

The Mean Absolute Error loss function avoids the problems with weighting outliers too strongly by scaling the loss only linearly. It is monotonically increasing, symmetric, homogenous and differentiable everywhere except at $e_{t+h} = 0$. Its linear nature would not make it compatible to many different problems as the losses produces might not be directly proportional to the errors in prediction.

Linex Loss Function

$$L(e_{t+h};\alpha_1,\alpha_2) = \alpha_1[e^{\alpha_2 e_{t+h}} - \alpha_2 e_{t+h} - 1], \quad \alpha_1 \geqslant 0, \alpha_2 \neq 0$$

Another commonly used loss function is the Linex loss function due to its asymmetric nature and differentiablity everywhere. Note that if $\alpha_2 > 0$, the function becomes almost linear to the left of the y-axis and almost exponential to the right. It flips around for $\alpha_2 < 0$. Any large under predictions, however, would be costlier than over predictions of the same magnitude if $\alpha_2 > 0$. The opposite effect applies if $\alpha_2 < 0$

Piecewise Asymmetric Loss Function Family

$$L(e_{t+h};a,b) = \begin{cases} aL_1(e_{t+h};p) & e_{t+h} > 0 \\ bL_2(e_{t+h;p}) & e_{t+h} < 0 \end{cases} \tag{4.12}$$

for $a, b > 0$ and p is a random variable.

We typically choose

$$L_2(t+t+h;p) = L_2(e_t+h,p) = |e_{t+h}|^p$$

Then, as special cases we have:

- $p = 1$: Lin-Lin case.
- $p = 2$: Quad-Quad case

Both are non-differentiable at zero, continuous and asymmetric if $a \neq b$, This family of loss functions would produce accurate losses for the errors at each time, however each of these expressions would need to be defined manually, which would be prone to error.

Hinge Loss Function

$$L = \max(0, e_{t+h}) \quad \text{or} \quad L = \max(0, e_{t_h})^2$$

where $e_{t+h} > 0$

This function is most suitable for one sided errors as it only produces losses for positive errors. For example, for $e_{t+h} = 7$, $L = \max(0, e_{t+h}) = 7$, therefore, a loss would be recorded. Conversely, if $e_{t+h} = -7$, $L = \max(o, e_{t+h}) = 0$, so no loss would be recorded. As large errors cause more serious losses, the function can be turned into a quadratic or even cubic functions to predict the losses more accurately.

We discuss the set up of the Hinge loss function to our model in the following section.

4.7.1 Hinge Loss Function Applied to Initial Economic Security Model

As described in the previous section, Hinge Loss functions are most suitable for one-sided errors. Since organisations would only be at risk if their current level of security is lower than their target level, and would only be producing a loss if they invested more than their target level of investment, we use Hinge Loss functions to capture this.

For example, a high level of confidentiality would only make an organisation's security system more secure and therefore not cause any damage, whereas a low level could cause damage. Conversely, high magnitudes of investments could cause the organisation to incur a loss, therefore, a loss would occur only when the level of investment exceeds its target.

With this in mind, we consider a one-sided loss function, where the loss only occurs when there is a lower level of security than the target or higher investment than initially planned.

Let $\bar{C}, \bar{I}, \bar{A}, \bar{\dot{K}}$ be the target levels of confidentiality, integrity, availability and change in investment, respectively. In this specification, a loss would only occur if $C < \bar{C}$, $I < \bar{I}$, $A < \bar{A}$, or $\dot{K} > \bar{\dot{K}}$.

Since a hinge loss function is 0 when no loss has occurred, all disturbances in the security levels are negative. Therefore, we take that all shocks are given as actual levels of the attributes below their targets and that investment expenditure is greater than planned. That is, $0 < \bar{C} - C$, $0 < \bar{I} - I$, $0 < \bar{A} - A$ and $0 < \dot{K} - \bar{\dot{K}}$.

We can then define a hinge loss function for the losses incurred by an organisation for each of the security attributes and change in investment in the following form:

- $L(C) = \max(0, \bar{C} - C)$
- $L(I) = \max(0, \bar{I} - I)$
- $L(A) = \max(0, \bar{A} - A)$
- $L(\dot{K}) = \max(0, \dot{K} - \bar{\dot{K}})$.

Following Granger(1999), we can add the loss functions for each element to construct one function for the total loss produced in a system as:

$$L(C, I, A, \dot{K}) = w_C L(C) + w_I L(I) + w_A L(A) + w_K L(\dot{K}) \tag{4.13}$$

where $\{w_C, w_I, w_A, w_K\}$ are the weights associated with each variable and sum to 1.

We can now use impulse-response functions to obtain a time profile of organisations when attacked. This concept is explained in detail in the next section.

4.8 Impulse-Response Functions

Impulse-response functions are used to measure the time profile of a single shock in a dynamic system. They can be thought of as the outcome of a thought experiment, where the time profile of the effect of a shock of size S hitting the system at time t is compared with the base-line profile at time $t + n$.

We can define an impulse-response function mathematically as:

$$\begin{aligned} I_y(n, \xi, \Omega_{t-1}) = \mathbb{E}[Y_{t+n} | S_t = \xi, S_{t+1} = 0, ..., S_{t+n} = 0, \Omega_{t-1}] \\ - \mathbb{E}[Y_{t+n} | S_t = 0, S_{t+1} = 0, ..., S_{t+n} = 0, \Omega_{t-1}] \end{aligned} \tag{4.14}$$

where Y_{t+n} is a dynamic function at time $t+n$, S_* represents the magnitude of the shock at time $*$, and Ω_{t-1} is the history of the economy up to time $t-1$ for $\xi > 0$ and $n = 1,2,3,...$ [108].

For the purposes of this thesis, we use impulse-response functions to describe the behaviour of organisations when one of their security attributes or change in investment is attacked. We let Y_t be any of the time evolutions of C, I, A, or $\dot{K}$ as defined in equations (4.2)-(4.5), with dynamic shocks as defined in equation (4.6).

Graphical illustrations of each of the time profiles are given in Section 4.11, where the model is applied to 4 different organisations as part of several thought experiments. In the next section, we define the stability conditions for our model to ensure its return to equilibrium at all times.

4.9 Stability Conditions

Stability conditions ensure the convergence of errors or pertubations over time, [102]. We assume that our system is inherently stable and once shocked will return, within a finite period, to its original state. To ensure this, we determine the range of parameter values for our system of interactions, as presented previously, based on the set of constraints required for the system to be stable on the Trace-Determinant plane. Here, a system is stable if it has a positive determinant and negative trace, [31, 75] for all t.

We start by re-writing our system of interactions in matrix form and use Mathematica to calculate its trace and determinant.

$$\begin{bmatrix} K \\ A \\ I \\ C \end{bmatrix} = \begin{bmatrix} -\chi & -\lambda & -\eta & 0 \\ \frac{-\epsilon\int|\dot{C}|dt}{\int\dot{C}dt} & -\psi & 0 & \delta \\ \sigma & 0 & w_{I_1}\phi_1 - w_{I_2}\phi_2 & -\iota \\ 0 & \tau & -\alpha & \beta \end{bmatrix} \begin{bmatrix} \int\dot{C}dt \\ \int\dot{I}dt \\ \int\dot{A}dt \\ \int\dot{K}dt \end{bmatrix} + \begin{bmatrix} \nu \\ \gamma \\ -\theta \\ -\omega \end{bmatrix} \int \dot{R}dt + \begin{bmatrix} 0 \\ 0 \\ I_0 \\ C_0 \end{bmatrix} - \begin{bmatrix} S_{K_t} \\ S_{A_t} \\ S_{I_t} \\ S_{C_t} \end{bmatrix} \tag{4.15}$$

Let $Parameters$ represent the square matrix of parameters in (4.15). We find that:

$$\begin{aligned} Det[Parameters] = &-\frac{\epsilon\int|\dot{C}|dt}{\int\dot{C}dt}(\alpha\iota\lambda + \eta\iota\tau + \beta\lambda(w_{I_1}\phi_1 - w_{I_2}\phi_2)) \\ &-(\eta\sigma - \chi(w_{I_1}\phi_1 - w_{I_2}\phi_2))(\delta\tau + \beta\psi) + \alpha(-\delta\lambda\sigma + \iota\chi\psi) \end{aligned} \tag{4.16}$$

and

$$Tr(Parameters) = \beta + w_{I_1}\phi_1 - w_{I_2}\phi_2 - \chi - \psi \tag{4.17}$$

For a negative trace, we ensure that the parameters satisfy the following condition for all t:

$$\beta + w_{I_1}\phi_1 < w_{I_2}\phi_2 + \chi + \psi$$

Therefore, we first set $\chi, \psi \gg 0$ and $\beta \to 0$ before moving on to determining the size of the parameters for a positive determinant.

The conditions for obtaining a positive determinant at all times is more complicated. Due to the evolving nature of $\int \dot{C} dt$ over time, we consider two cases; (i) $\int \dot{C} dt > 0$ and (ii) $\int \dot{C} dt < 0$, and estimate the parameters accordingly.

Case (i)

Given that $\int \dot{C} dt > 0$, the determinant, (4.16), simplifies to:

$$\begin{aligned} Det[Parameters] = &-\epsilon(\alpha\iota\lambda + \eta\iota\tau + \beta\lambda(w_{I_1}\phi_1 - w_{I_2}\phi_2)) \\ &- (\eta\sigma - \chi(w_{I_1}\phi_1 - w_{I_2}\phi_2))(\delta\tau\ + \beta\psi) + \alpha(-\delta\lambda\sigma + \iota\chi\psi) \end{aligned}$$

Let us consider the parameters within the initial set of parentheses first:

$$-\epsilon(\alpha\iota\lambda + \eta\iota\tau + \beta\lambda(w_{I_1}\phi_1 - w_{I_2}\phi_2))$$

Here, we would require $\epsilon, \beta, \lambda \to 0$ and $w_{I_1}\phi_1 > w_{I_2}\phi_2$, to obtain a very small negative value overall.

In the second set of brackets,

$$-(\eta\sigma - \chi(w_{I_1}\phi_1 - w_{I_2}\phi_2)),$$

we let $\eta, \sigma \to 0$ whilst $\chi \gg 0$ to ensure a negative outcome within the bracket, which will turn positive when multiplied by the negative 1 outside it. This is then multiplied by the third bracket,

$$(\delta\tau\ + \beta\psi),$$

where $\tau \gg 0$, to form a large, positive value overall.

Since the outcome of the initial bracket was very small and negative, and the second bracket was large but positive, we would now have an overall positive value. In order to ensure that this positivity is maintained, we set the parameters in the final set of parentheses so that they produce a positive outcome. Therefore,

$$\alpha(-\delta\lambda\sigma + \iota\chi\psi) > 0$$

if $\iota \gg 0$ and $\delta \to 0$, to ensure that α multiplies a positive number, where $\alpha > 0$.

These conditions allow us to ensure a positive determinant when $\int \dot{C}dt > 0$.

Let us now consider the second case.

Case (ii)

Given that $\int \dot{C}dt < 0$, the determinant, (4.16), simplifies to:

$$\begin{aligned} Det[Parameters] &= \epsilon(\alpha\iota\lambda + \eta\iota\tau + \beta\lambda(w_{I_1}\phi_1 - w_{I_2}\phi_2)) \\ &\quad - (\eta\sigma - \chi(w_{I_1}\phi_1 - w_{I_2}\phi_2))(\delta\tau \ + \beta\psi) + \alpha(-\delta\lambda\sigma + \iota\chi\psi) \end{aligned}$$

In this case, for the first set of parenthesis,

$$\epsilon(\alpha\iota\lambda + \eta\iota\tau + \beta\lambda(w_{I_1}\phi_1 - w_{I_2}\phi_2)),$$

we would now only require $\epsilon \gg 0$ and $w_{I_1}\phi_1 > w_{I_2}\phi_2$ to ensure that it remains positive.

For the second and third set of parentheses,

$$-(\eta\sigma - \chi(w_{I_1}\phi_1 - w_{I_2}\phi_2))(\delta\tau \ + \beta\psi),$$

we, again, let $\eta, \sigma \to 0$, whilst $\chi, \tau \gg 0$, to obtain a positive value.

Finally, for the last set of brackets,

$$\alpha(-\delta\lambda\sigma + \iota\chi\psi),$$

we let $\delta, \lambda \to 0$ whilst $\iota \gg 0$ to ensure a positive final bracket.

The sum of these values would then give a positive determinant, when $\int \dot{C}dt < 0$.

Note that, we cannot identify one unique set of parameters which would satisfy all the conditions, as they can be satisfied with several different combinations. Instead, we ensure that the parameters used in the system depicting interactions and those used in the subsequent calculations of the impulse-response functions, satisfy these conditions at all times.

In the following section, we carry out a sensitivity analysis for the remaining variables to determine the set of values for which the system remains valid.

4.10 Sensitivity Analysis

Sensitivity analysis is a technique which is used to determine how different values, within a specific boundary, of an independent variable would impact a particular dependent variable, under a given set of assumptions. It is used as a method for predicting the outcome of decisions and aids the decision maker in taking informed and appropriate decisions.

Monte Carlo simulations are often used to explore the range of values for which the variables produce valid results. The range of input values are taken from a probability distribution, such as uniform or normal distributions, and results are calculated for hundreds or thousands of different values. These are then used to produce distributions of possible outcome values.

Since we have already calculated the range of parameter values which can be used to ensure the stability of the system in the previous section, we will only perform a sensitivity analysis on the range of target values as the system responds to the deviations from target.

We use Monte Carlo simulations to determine the range of values these targets can take to produce valid results, in the form of impulse-response functions.

We start by setting up a hypothetical system which has equal preferences in each of the security attributes in order to negate the effect of system preferences on our results. That is, $w_C = w_I = w_A = w_K = 0.25$.

We let each of the security attributes target values take values between 0 and 100 uniformly, and plot their respective impulse-response functions using MATLAB. This process is repeated 500 times to observe the effect of different target values on the model. We find that all values used return the system to equilibrium within 30 time steps, thus confirming the validity and robustness of the model.

These results are illustrated in Figures 4.1 – 4.4.

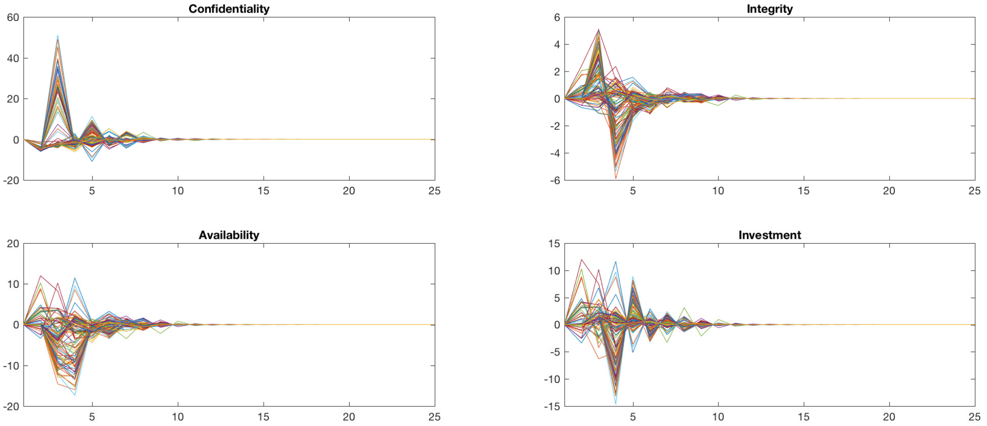


Figure 4.1: Impulse-response function for various target values of confidentiality

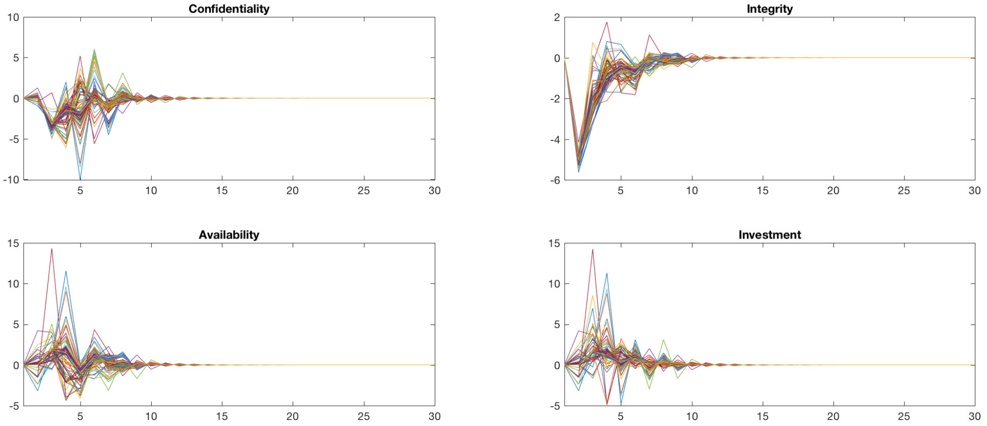


Figure 4.2: Impulse-response function for various target values of integrity

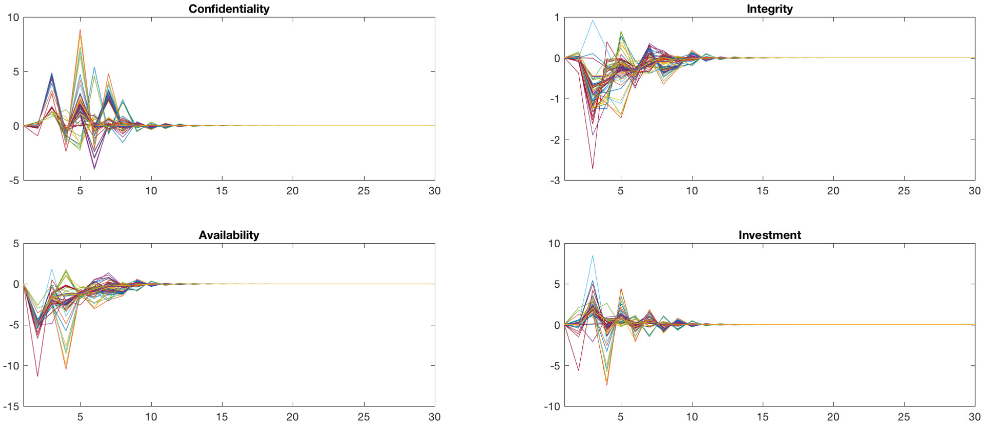


Figure 4.3: Impulse-response function for various target values of availability

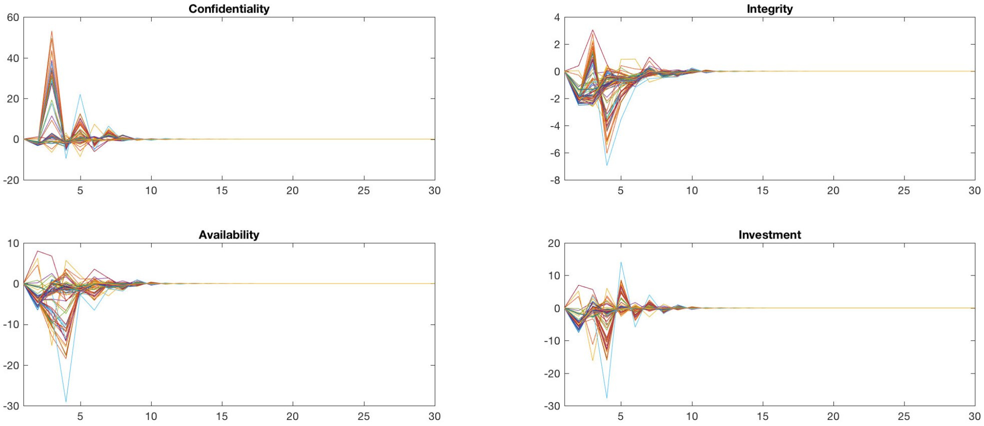


Figure 4.4: Impulse-response function for various target values of investments made

4.11 Thought Experiment Example Organisations

Due to the lack of data, we apply our model to 4 different organisations as part of a series of thought experiments to show how the model behaves in different situations. As stated by Brown and Fehige (2014), 'thought experiments are devices of the imagination used to investigate the nature of things'. We use these to assess our model's validity and reliability by applying it to medical research organisations, retail, bank and military weapons systems. We discuss the nature of each of the organisations with respect to their preferences in the security attributes and investment and then observe their behaviour, in the presence of shocks to each of the security attributes and investment separately. A summary of each of their preferences is provided in Table 4.1.

Organisation Type	**Preference in $C, I, A, \dot{K}$**	**Preference in (non-)availability**
Medical Research	$w_I > w_C > w_A > w_K$	$w_{I_1} < w_{I_2}$
Retail	$w_A \simeq w_K > w_I > w_C$	$w_{I_1} \gg w_{I_2}$
Bank	$w_C \simeq w_I > w_A > w_K$	$w_{I_1} < w_{I_2}$
Military Weapons System	$w_A \gg w_I > w_C > w_K$	$w_{I_1} \gg w_{I_2}$

Table 4.1: Organisational preferences

Note that, we let each of the organisations have the same parameter values (conforming to the conditions described in Section 4.9) as we are only interested in their behaviour over time based on their preferences and type of attack they are facing.

At the end of this section, we also analyse the total levels of investments made in security measures for attacks on each of the security attribute over time, to further explore the dimensions of our model.

We begin by analysing the behaviour of a medical research organisation in the next section.

4.11.1 Medical Research Organisation

In order to ensure the accuracy and reliability of their research, a medical research organisation would prioritise the integrity of their data the most. This would be closely followed by the level of confidentiality of their data to ensure that only authorised personnel are able to view the data.

The availability of their resources, such as any research equipment, would also be important to them followed by the investments made in the organisation. Therefore, we set $w_I > w_C > w_A > w_K$.

In this case, the integrity of the data would be maintained through limiting its availability, therefore we also set $w_{I_1} < w_{I_2}$.

Using MATLAB, we produce the impulse-response functions for a medical research organisations with the given set of preferences. These are illustrated in Figure 4.5 and 4.6.

Figure 4.5 shows the behaviour of the organisation when the confidentiality or integrity is attacked. We observe that an attack on the integrity of their data also causes a drop in its confidentiality as the attacker would have been able to damage their data by gaining access to it in the first place. This would prompt the organisation to decrease its connectivity between location, to make it more difficult for the attacker to move between them, whilst simultaneously increasing investments in security measures to aid in restoring the system back to equilibrium.

We can see that attacks on the confidentiality of the system also cause the organisation to initially decrease its connectivity as well as the availability of its services to prevent the attacker from accessing them. Investments are then being made in the security measures to restore the system back to equilibrium over time.

The behaviour of a medical research organisation in the presence of attacks to the availability and investment are illustrated in Figure 4.6.

Here, attacks on the availability of the system causes the system managers to decrease the complexity of their system in order to avoid movement of attacks and increase the level of confidentiality to restrict access for any further attacks. The system is able to restore itself back to equilibrium over time by making investments and increasing the level of each of the security attributes.

Attacks on the investment, however, initially cause a decline in the levels of all three attributes, since the attacker would have been able to gain access to their finances, take some, and therefore make it less available to be used. The security manager then overcomes this attack by reducing the complexity of the system and making it difficult for the attacker to move around in the system, to restore it back to its equilibrium state over time.

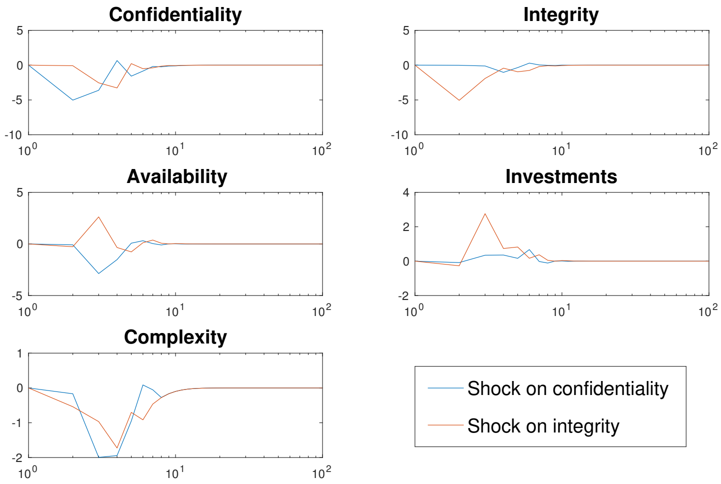


Figure 4.5: Impulse-response function for a medical research organisation when its protection of the confidentiality and integrity of their data has been compromised.

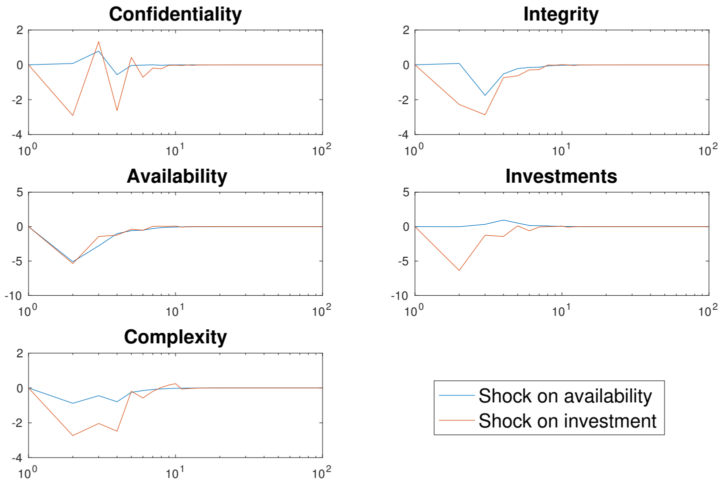


Figure 4.6: Impulse-response function for a medical research organisation when its protection of the availability and investment of their data has been compromised.

4.11.2 Retail Organisation

Retail organisations would tend to be most concerned about the availability of their products in stores and the amount they invest in their business, due to their business objective of sales and profit maximisation. This would mean that they would tend to prioritise the availability of their product the most as well as their investments. Therefore, we set $w_A \simeq w_K$ to capture this preference. These organisations would then be most concerned about the integrity of their products, (i.e. ensure that they are as advertised, not faulty or damaged in any way) followed by the level of confidentiality in their shops to ensure, for example, that customers don't enter the storage rooms, etc. Hence, we set the preferences to be $w_A \simeq w_K > w_I > w_C$

In this case, a high availability in the products would help to have a higher level of integrity as it would reduce the likelihood of selling a faulty item. Therefore, we set $w_{I_1} \gg w_{I_2}$.

The behaviour of a retail organisations when each of its security attributes or level of investment is attacked is shown in Figure 4.7 and 4.8 in the form of impulse response functions.

We observe that a retail organisation, when faced with attacks on their confidentiality or integrity, responds by increasing their investments in security measures whilst reducing the complexity of the system, to prevent the attacker from easily moving around within the system. In the case of an attack on the confidentiality of the system, the system manager's respond by reducing the availability of their services to prevent any further such attacks. However, the availability of their services is increased for an attack on the integrity, since the integrity is maintained through the availability of their products in this example. In both cases, the system is able to restore itself back to equilibrium over time.

We can see from Figure 4.8 that an attack on the availability of the products in the retail organisation would also cause a drop in its integrity levels. Again, this is due the integrity of the products being maintained through their availability. If, for example, there aren't many products available for sale, the chances of selling a damaged item increases, therefore reducing their integrity. Attacks on the investment of the retail organisation cause drop in all three security attributes and force the security manager to reduce its connectivities between locations and slowly increase it again to allow the system to go back to equilibrium.

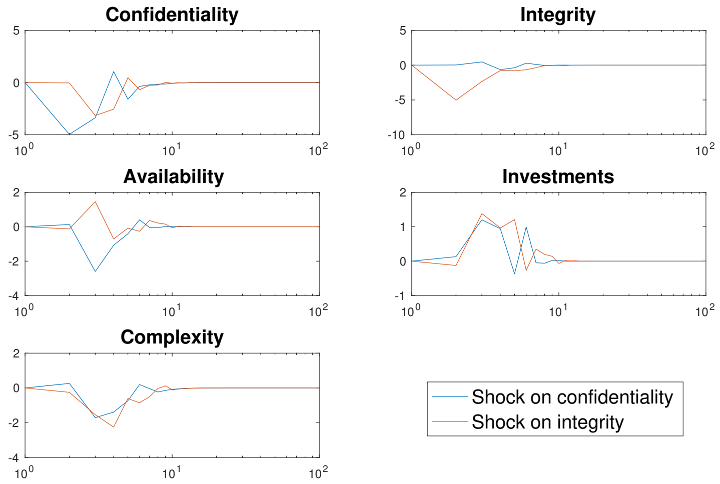


Figure 4.7: Impulse-response function for a retail organisation when its protection of the confidentiality and integrity of their data has been compromised.

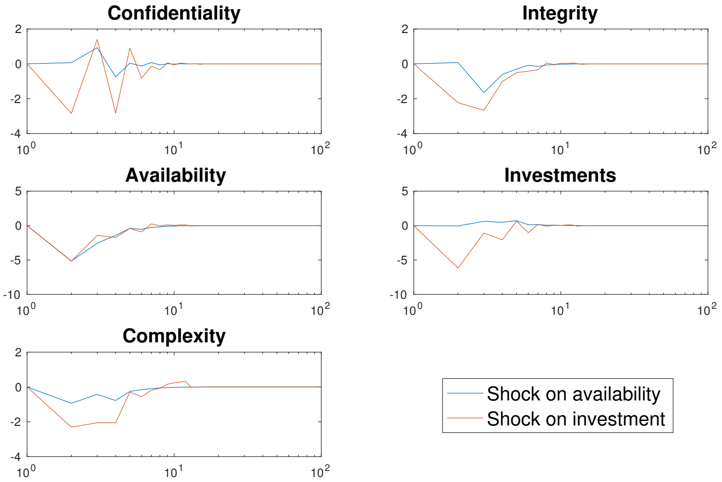


Figure 4.8: Impulse-response function for a retail organisation when its protection of the availability and investment of their data has been compromised.

4.11.3 Bank

A bank would be most concerned about the confidentiality and the integrity of their customer's data more than the availability of their services or the investment they make. They would require the data that they hold for their customers to be accurate, when lending loans for example, and confidential, to ensure the customers' personal and financial details stay private. Since these two attributes are similar in their importance in this case, we let $w_C \simeq w_I$. This is followed closely by high levels of availability to ensure that customers are able to gain access to their data/accounts when required, followed by the investments they make so $w_A > w_K$. Hence, the preferences are set to be $w_C \simeq w_I > w_A > w_K$.

In this case, the protection of the integrity of their data would be maintained through its non-availability, therefore we set $w_{I_1} < w_{I_2}$.

The impulse-response functions when each of the attributes or investment is attacked is shown in Figure 4.9 and 4.10, where, in all cases, the system is able to restore itself back to its equilibrium position over time.

We find that the security manager tries to overcome the attacks by reducing the system's complexity for all cases, except for attacks on availability. In this case, they increase the complexity of the system to, for example, run back up services, until the system is able to restore itself back to equilibrium, in which case the complexity of the system also goes back to its original state. This is illustrated in Figure 4.10.

We can see from Figure 4.9 that attacks on the integrity or confidentiality of the system initially cause a drop in the level of investments, as the attacker could now also have gained access to a bank's finances. However, the system's manager responds to such attack by investing in security measures rapidly and increasing the levels of all security attributes, to restore the system back to its equilibrium position over time.

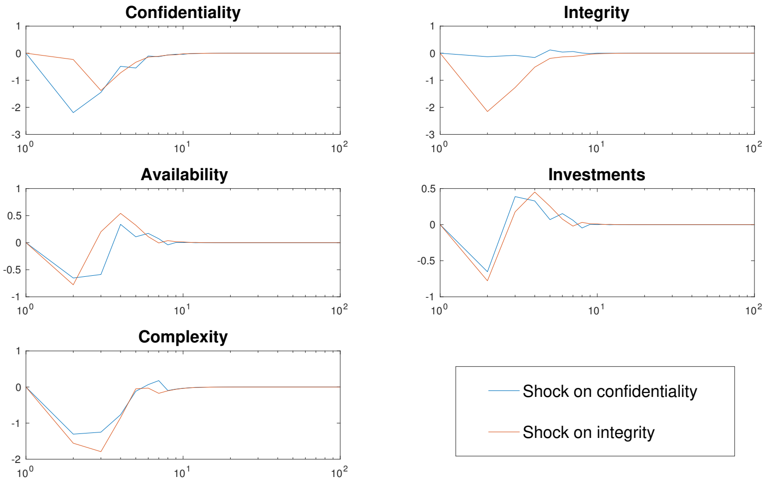


Figure 4.9: Impulse-response function for a bank when its protection of the confidentiality and integrity of their data has been compromised.

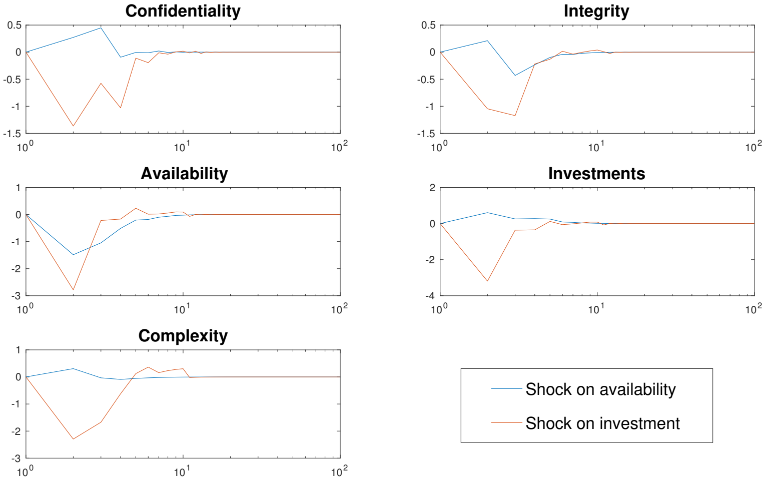


Figure 4.10: Impulse-response function for a bank when its protection of the availability and investment of their data has been compromised.

4.11.4 Military Weapons System

The main aim of the operators of a military weapons system would be to defeat their opponents with the weapons available, and the information that they have been given regarding them. Therefore, they would prioritise the availability of their weapons the most, to ensure that they are never short on supplies in the middle of operations. After this, they would be most concerned with the integrity of their weapons, to ensure that they are not faulty, followed closely by their levels of confidentiality to prevent any unauthorised access to them. They would be less concerned about the costs involved with obtaining new weapons. Therefore, we set the manager's preferences to be $w_A \gg w_I > w_C > w_K$

Furthermore, the integrity of their weapons system would be maintained through the availability of weapons since this would reduce the possibility of a faulty weapon in their midst. Therefore, we set $w_{I_1} \gg w_{I_2}$ to ensure that there are enough fully functioning weapons available when needed.

The impulse-response functions for the behaviour of military weapon systems when each of the levels of its security attributes or investment is attacked, is given in Figure 4.11 and 4.12.

We find that, similar to a bank, the military weapon's system's initial reaction to these attacks would be to reduce its complexity for all types of attacks except availability, since a reduction in the availability of their weapons could potentially cause devastating consequences, Therefore, thye would need to overcome this type of attack by increasing their weapons supply through increased investments, which would increase the complexity of the system. This is illustrated in Figure 4.12.

We find that the system is able to restore itself back to equilibrium over time for all 4 attacks.

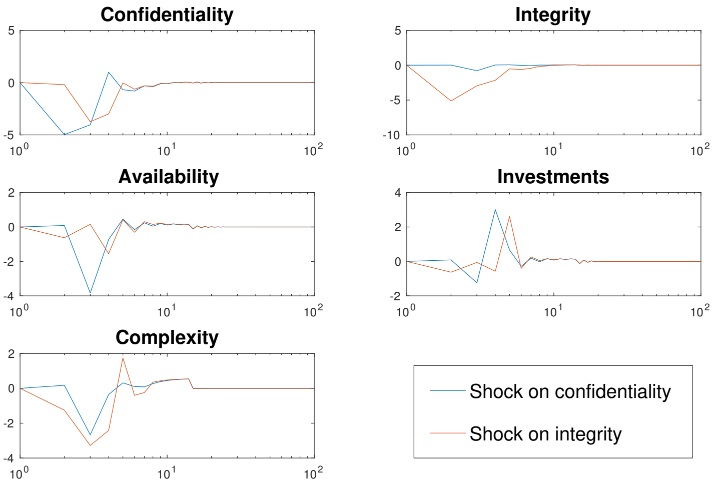


Figure 4.11: Impulse-response function for a military weapons system when its protection of the confidentiality and integrity of their data has been compromised.

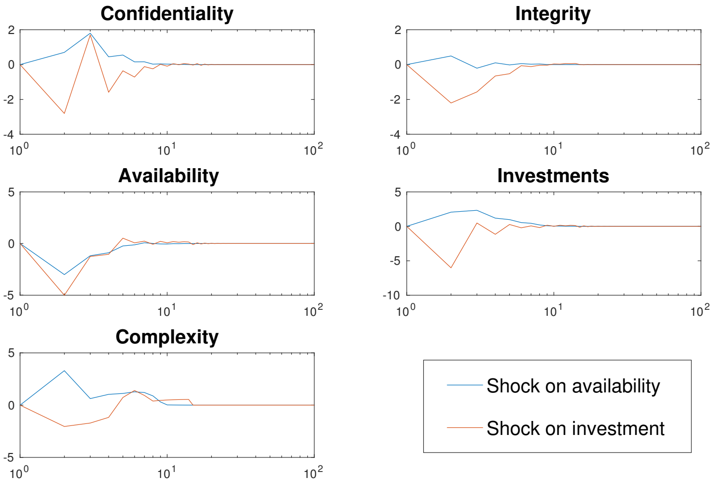


Figure 4.12: Impulse-response function for a military weapons system when its protection of the availability and investment of their data has been compromised.

The total level of investments made for each of the organisations in the presence of attacks to each of their security attributes is given in Table 4.2.

We find that the magnitude of the level of investments made correspond to the preferences in each of the security attributes for each organisation. A medical research organisation and bank, for example, invested the most when faced with an attack on the integrity of their data whereas a retail organisation and military weapons system invested the most when the the availability of their resources was being compromised.

These results aid in further validating our model by confirming that organisations would tend to invest the most in attacks exploiting their most valued security attribute in their organisation.

Type of Organisation	**C**	**I**	**A**
Medical Research	1.3243	4.7108	2.2912
Retail	0.6647	1.2866	1.9807
Bank	1.3971	2.2514	1.0664
Military Weapons System	3.1046	1.9855	7.9823

Table 4.2: Total value of investments made over time for attacks on each of the security attributes

5 MATLAB

The MATLAB implementation for this model consists of 4 different files: `stateDynamics.m`, `lossFunction.m`, `optimizationExpectedUtility.m` and `runningFile.m`. The next bullet points explain each of the files used and their roles in the code.

- `stateDynamics.m`

  This file sets up a function whose four outputs are the levels of C, I, A and $\dot{K}$, as defined in equations (4.7) – (4.10), with their respective components and shocks forming the inputs. Here, `Cnew`, `Inew`, `Anew`, `K_change` are the levels of C, I, A and $\dot{K}$ and `C`, `I`, `A`, `K` represent $\dot{C}$, $\dot{I}$, $\dot{A}$ and $\dot{K}$, respectively.

  This file will be used later in the `lossFunction.m` and `optimizationExpectedUtility.m` file to aid in finding the optimal values of `Cnew`, `Inew`, `Anew`, and `K_change` over time.

- `lossFunction.m`

  The `lossFunction.m` file describes the loss function as in (4.13). Its inputs include the outputs from `stateDynamics.m`, the already-set targets as chosen by the security manager, `C_target`, `I_target`, `A_target` and `K_target`, and the weights for each deviation, `wC`, `wI`, `wA` and `wK`.

- `optimizationExpectedUtility.m`

  This file optimizes the loss function to find the levels of `Cnew`, `Inew`, `Anew` and `K_change` that produce the minimum amount of loss over time. Its input values are the joint input values of the `stateDynamics.m` and `lossFunction.m` files in addition to `S`, which is taken to be the amount of time an agent can see ahead in the future and `DF`, which is the discount factor.

  The code starts by defining some variables to be used. `xCvec`, `xIvec`, `xAec`, `xKec` are the control variables to determine the complexity of the system over time as defined in equation 4.11. These are set to be linearly spaced vectors taking values between `eps` and `1-eps`, where `eps` is a very small number close to `0`.

  Next, two 4 dimensional matrices are defined as `Loss` and `R_vals` to capture the losses and the system's response to deviations in terms of C, I, A and $\dot{K}$, respectively.

  We then set up a nested for-loop, from `i=1 - NUM` (`NUM` is any integer), and find the various possible values of the complexity, $C, I, A, \dot{K}$ and $Loss$ given the changes in each of the control variables. The total losses for each of the attributes and investment are then added together and recorded in the Loss matrix as previously defined.

The combination of control variables producing the minimum loss over time is then found and used to find the corresponding values of complexity, security attributes and investment. We take these to be the optimal levels of C, I, A and $\dot{K}$ over time.

- runningFile.m

  This file produces 6 graphs which capture the behaviour of the levels of C, I, A and K when the system is being shocked, in the form of impulse-response functions, and complexity, R, of the system.

  It starts by setting appropriate values for the weights, target values, initial values of integrity and confidentiality and the persistence of shocks to be added, all corresponding to the type of organisation to be considered.

  A for-loop is then created where the parameter values are set according to the conditions discussed previously for when $\int_{t_0}^{t} \dot{C} dt < 0$ and $\int_{t_0}^{t} \dot{C} dt < 0$. The magnitude of the shock on the confidentiality of the system is also determined in this for-loop. The loop starts each of the attributes and investment at 0, and adds shocks at i=2 by calling back the optimizationExpectedutility function. After the initial addition of the shock, the system continues without any further addition of the shock, allowing it to disintegrate over time. The behaviour of the attributes and investment over time are then recorded in the Z matrix whose columns are defined as follows:

  - Column 1 represents the values of x which is the control value that produces the least losses for C at time i,
  - Column 2 represents the C values, which are the optimal levels of confidentiality at time i,
  - Column 3 represents the I values, which are the optimal levels of integrity at time i,
  - Column 4 represents the A values, which are the optimal levels of availability at time i,
  - Column 5 represents the "K"values, which are the optimal levels of change in investment at time i,
  - Column 6 represents the "R"values, which is the complexity of the system that produced the optimal levels of C, I, A and K at time t,

- Column 7 represents the values of y which is the control value that produces the least losses for I at time i,
- Column 8 represents the values of z which is the control value that produces the least losses for A at time i, and
- Column 9 represents the values of l which is the control value that produces the least losses for K at time i.

We then observe the behaviour of the system, without the presence of any shocks, similar to above, and record the values obtained in the matrix Z0, which takes the same form as Z.

In order to find the impulse-response function for this system, we subtract each element of Z0 from its respective element in Z, as explained in the definition of impulse-response functions above. This produces our first impulse-response function for C, I, A, K and R referred to as Impulse1

This process is repeated an additional 3 times for the same organisations, but with shocks added to each of integrity, availability or investment separately. This produces the matrices Impulse2, Impulse3 and Impulse4.

Each of the impulse-response functions are then plotted on graphs to compare the behaviour of an organisation to different types of attack over time.

6 Enriched Economic Security Model

Now that we have introduced the initial economic security model and confirmed its validity and robustness, we can incorporate the integration of the distributed systems framework into it. This can later be used to obtain a better understanding of the movement of an attacker through the system, and how much damage it causes at each location over time.

In this section, we concentrate on the integration of the distributed systems framework with the economic security model only. Due to the current lack of dynamics in this model, (i.e. it does not incorporate the effect of attacks on a system, rather it only provides an representation of the different abstractions present in an organisation's security system) we will only concentrate on explaining the changes and additions that this integration brings to the initial economic security model. The incorporation of an attacker model into this is discussed in Section 8 as a recommendation for future work.

We begin by introducing a matrix environment to the current set of system evolutions as described in equations (4.7) – (4.10). These show the levels of protection placed to maintain the confidentiality, integrity and availability of resources at location i from attacks from location j, and the investments required to maintain these at each location.

We let C_{ij}, I_{ij}, and A_{ij} be the levels of protection of the confidentiality, integrity, and availability of the resources at location i from attacks from location j, and $\dot{C}_{ij}$, $\dot{I}_{ij}$, and $\dot{A}_{ij}$ be their respective rates of change over time. K_{ij} is taken to be the level of security investments made to protect location i from attacks from location j, and $\dot{K}_{ij}$ is the rate of change of this investment over time.

Note that, we are now representing the properties of the resources at each location in terms the three security attributes. The notion of processes is now captured as the evolution of the model over time, and their environment is taken to be the system of organisations including all the locations within it.

In the following sections, we introduce the idea of a placeholder for attacks in the model. We then move on to describe the trade-offs between the security attributes and investment over time whilst also incorporating the network model. Similar to their relationships in the previous section, these will be based on a set of assumption, which will be outlined in their respective sections.

Note that, although there may exist an organisation which does not behave in the way we have defined the time evolutions to be, we aim to make these representations as general as possible within the bounds of the given assumptions.

6.0.1 Placeholder (R)

In order to incorporate the attack model in future work, we now let the matrix $R_{\star_{ij}}$, where $\star = \{C, I, A, \dot{K}\}$ and $r_{\star_{ij}} \in R_{\star}$, be a placeholder which represents the pay-off of an attack that is able to propagate itself within the system, with respect to the confidentiality, integrity, availability or investments made in the system.

We can use the stochastic Markov models, as laid out in Section 7, to obtain the optimal strategy for the attacker at each state, in order to find the expected pay-off of the attacker at each location. This pay-off will be respective of the damages caused to each location in terms of the confidentiality, integrity, availability of the system and investments required to restore it. That is, the elements in the matrix R will be dependent on the damages caused to the security attributes and investment levels of the system.

We assume that the pay-off of the attacker will have a negative effect on the protection of the investment, confidentiality, integrity and availability of the system, since an attacker's aim would be to damage the system.

For example, the level of protection of the integrity of a sensitive document would have decreased if the document has been damaged. Similarly, the level of protection of the availability of a system would also decline if a denial-of-service attack was able to by-pass the controls and disrupt any services provided.

6.0.2 Security Attributes and Investment

Our new system of equations, with the incorporation of the network model, now become:

$$C_{ij} = -\alpha \int_{t_0}^{t} \dot{A}_{ij} dt + \beta \int_{t_0}^{t'} \dot{K}_{ij} dt + \tau \int_{t_0}^{t'} \dot{I}_{ij} dt' + C_{0_{ij}} - R_{C_{ij}} \tag{6.1}$$

$$I_{ij} = \sigma \int_{t_0}^{t'} \dot{C}_{ij} dt' + \iota \int_{t_0}^{t'} \dot{K}_{ij} dt' + (w_{I_1}(\phi_1) - w_{I_2}(\phi_2)) \int_{t_0}^{t'} \dot{A}_{ij} dt' + I_0 - R_{I_{ij}} \tag{6.2}$$

$$A_{ij} = \delta \int_{t_0}^{t'} \dot{K}_{ij} dt' - \epsilon \int_{t_0}^{t'} |\dot{C}_{ij}| dt' - \psi \int_{t_0}^{t} \dot{I}_{ij} dt' + I_{0_{ij}} - R_{A_{ij}} \tag{6.3}$$

$$\dot{K}_{ij} = -(\chi \dot{C}_{ij} + \lambda \dot{I}_{ij} + \eta \dot{A}_{ij} + R_{K_{ij}}) \tag{6.4}$$

where each of the parameters, again, describe the impact of their respective components to the security attributes or change in investment, and $\{w_1, w_2\}$ are the weights placed on the availability or non-availability of the resources at each location, respectively, in order to preserve the integrity.

6.0.3 Loss Function

We break down the notation of the loss function to capture the losses at each location as follows:

$$L_{ij}(C_{ij}, I_{ij}, A_{ij}, \dot{K}_{ij}) = w_{C_{ij}} L(C_{ij}) + w_{I_{ij}} L(I_{ij}) + w_{A_{ij}} L(A_{ij}) + w_{K_{ij}} L(\dot{K}_{ij}) \tag{6.5}$$

where $\{w_{C_{ij}}, w_{I_{ij}}, w_{A_{ij}}, w_{K_{ij}}\}$ are the weights associated with respect to each variable at each locations and sum to 1 for each location.

Each $L(\star_{ij})$, where $\star = \{C, I, A, \dot{K}\}$, shows the losses produced at each location in the form of a hinge loss function as described previously. That is,

- $L(C_{ij}) = \max(0, \bar{C}_{ij} - C_{ij})$
- $L(I_{ij}) = \max(0, \bar{I}_{ij} - I_{ij})$
- $L(A_{ij}) = \max(0, \bar{A}_{ij} - A_{ij})$
- $L(\dot{K}_{ij}) = \max(0, \dot{K}_{ij} - \bar{\dot{K}}_{ij})$.

Again, we assume that the organisations makes a loss only when the level of confidentiality, integrity or availability at a location is less than the target level, or if it invests more than its target value of investment for a certain location.

6.0.4 Stability Conditions

We ensure the stability of our model by carrying out similar computations as in Section 4.9. We set out our system of equations as follows:

$$\begin{bmatrix} K_{ij} \\ A_{ij} \\ I_{ij} \\ C_{ij} \end{bmatrix} = \begin{bmatrix} -\chi & -\lambda & -\eta & 0 \\ \frac{-\epsilon \int |\dot{C}| dt}{\int \dot{C} dt} & -\psi & 0 & \delta \\ \sigma & 0 & w_{I_1}\phi_1 - w_{I_2}\phi_2 & -\iota \\ 0 & \tau & -\alpha & \beta \end{bmatrix} \begin{bmatrix} \int \dot{C}_{ij} dt \\ \int \dot{I}_{ij} dt \\ \int \dot{A}_{ij} dt \\ \int \dot{K}_{ij} dt \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ I_{0_{ij}} \\ C_{0_{ij}} \end{bmatrix} - \begin{bmatrix} R_{K_{ij}} \\ R_{A_{ij}} \\ R_{I_{ij}} \\ R_{C_{ij}} \end{bmatrix} \tag{6.6}$$

We find that, the change in the level of abstraction of the model does not affect the stability conditions since the determinant and trace remain roughly the same. That is,

$$Tr(Parameters) = \beta + w_{I_1}\phi_1 - w_{I_2}\phi_2 - \chi - \psi \tag{6.7}$$

and

$$
\begin{aligned}
Det[Parameters] = &-\frac{\epsilon\int|\dot{C}_{ij}|dt}{\int\dot{C}_{ij}dt}(\alpha\iota\lambda+\eta\iota\tau+\beta\lambda(w_{I_1}\phi_1-w_{I_2}\phi_2)) \\
&-(\eta\sigma-\chi(w_{I_1}\phi_1-w_{I_2}\phi_2))(\delta\tau\ +\beta\psi)+\alpha(-\delta\lambda\sigma+\iota\chi\psi)
\end{aligned}
\tag{6.8}
$$

where $Parameters$ is the matrix of parameters as shown in equation (6.6).

We can see that the trace has not changed, and the determinant now incorporates a more abstract representation of the level of confidentiality of the system. Again, we can split the determinant of the system into two cases, (i) $\int\dot{C}_{ij}dt > 0$ or (ii) $\int\dot{C}_{ij}dt < 0$ to obtain similar observations as in Section 4.9. To avoid repetition, we will not discuss those observations again in this section.

Because of the lack of dynamics in this system (absence of attack model does not allow for any evolutions in the system), we are unable to perform a sensitivity analysis to observe the effect of change in variables on the model. However, a sensitivity analysis can be performed in future work when an attack model has been incorporated.

The next section provides a suggested attacker model which, as well as creating dynamics in this model, it would allow for the observation of the movement of the attacker through a network and the damages it would cause at each location.

7 Attacker Model

Now that we have shown how a system could be represented in a distributed systems framework embedded within an economic model, it is important to understand how attackers could move throughout a system presented in this way. We set up an attacker model to capture this movement and the damages it causes to the system at each location. We then use this model to find the optimal strategies for the defender and attackers and their expected losses/pay-off at each stage. More details on this are given in Section 7.5.

We start this section by providing the reader with some definitions of some commonly used terms when talking about attacks, such as vulnerabilities, exploits and threats in Section 7.1.

We then give examples of different types of current attacks, outlining their aims, consequences and ways to defend against such attacks in Section 7.2. This, of course, is not a complete list of different types of attacks present, however, we believe it provides the reader with a starting point to understand the various motives and effects an attacker could have on different systems.

Section 7.3 describes the general Markov Decision Processes model, which is used to track the movement of agents in a system. We use this to model the movement of attackers in a system in the distributed systems framework in Section 7.4.

After this, we explain the notion of Stochastic Markov Security Games and Linear Influence Models in Sections 7.5 and 7.6, respectively. The integration of these will allow us to model the optimal strategy of an attacker and defender in a given interconnected system, and the expected loss produced for the defender/pay-off of the attacker in Section 7.7.

A numerical example of this integration is given in Section 7.7.1 to aid the reader in his understanding of the model.

Our main contributions in this section lie in the integration of the Markov Decision Process with the distributed system framework in Section 7.4. We have found from the explored literature that (Partially Observable) Markov Decision Processes are often used to describe the decisions a defender would take when a system is attacked. We, therefore, aim to apply this framework to represent the attacker instead, and observe its interaction with the defender in a zero-sum Markov security game as presented in Section 7.7.

A summary of the notations used throughout this section is given in the Appendix in the form of tables, for the reader's convenience.

7.1 Definitions

We provide the reader with some useful definitions of commonly used terms when discussing attacks, which we will use throughout this thesis.

- **Vulnerabilities:**

  Vulnerabilities are unintended flaws or gaps in a system which allow for the possibility of attack in the form of unauthorised access or malicious behaviour, such as viruses or worms. These can be caused by weak passwords, software bugs or computer viruses, which will need to get fixed through, for example, patching, or the requirement of passwords to contain a variety of capital letters, symbols and numbers, [93, 178].

  Security industries, cybercriminals and other individual are constantly in search of vulnerabilities, either to protect the system or to attack it. Once a vulnerability is found, there is a discussion on how much information should be disclosed to the public. Some believe a full and immediate disclosure should be made, including the specific information that could be used to exploit the vulnerability. Others believe that this information should not be published at all to avoid getting attacked. For example, a zero-day exploit occurs as soon as a vulnerability becomes known.

  Many experts, however, believe that limited information should be made available to only a selected group after some specified amount of time after detection to mitigate the risk of getting attacked, [149].

- **Exploits**

  An exploit is an attack on a computer system, usually performed by taking advantage of a particular vulnerability in an operating system or vended program. Used as a verb, the term refers to the act of successfully making such an attack.

  System managers overcome this attack by issuing a patch, which the users of the system will have to obtain themselves by either downloading it from the software developer or the web, or it could be downloaded automatically by the application that needs it. If the user decides not to install the patch, they expose themselves to a possible security breach.

  Some types of exploits include SQL injection attacks, zero-day exploits, and cross-site request forgery. Exploits can occur through a variety of ways, such as through malicious websites or the victim clicking on a link in a phishing email, [142].

- **Threats**

  A threat is a potential for a vulnerability to turn into an attack on the computer system, network etc., and potentially cause serious harm. These can put individual or business' computers at risk and will need to be fixed quickly to prevent attackers from infiltrating the system and causing damage.

  Threats include anything from viruses and trojans to attacks from hackers. The term 'blended threat' is often seen to be a more accurate as the threats often involve more than one exploit. For example, a hacker may use a phishing attack to gain information about a network and break into it, [169].

- **Attack**

  An attack includes any attempt to destroy, expose, alter, disable, steal or gain unauthorized access to or make unauthorized use of an asset, [184]. It is one of the biggest information security threat which can happen to both individuals and organisations. Some examples of attacks include botnet, phishing, DDoS or theft.

  Attacks can be separated into passive and active attacks. Passive attacks are mainly concerned about obtaining data but do not affect the system, such as wiretapping, whereas active attacks, could potentially cause major damage to an individual's or organization's resources as they aim to alter system resources or affect how they work, such as a virus, [170].

7.2 Different Types of Attack Vectors

Now that we have defined some common terms, we look at some of the various types of attack vectors in this section. Note that this is not an exhaustive list of all the different types of attacks that are present, rather, it is to provide the reader with an understanding of the various attackers' aims, consequences of attack and ways to defend yourself against them.

The analysis of these will help us to understand how they enter the system and the factors they take into consideration when deciding on the next place of exploit in a system. This will aid in determining the parameter values in the Markov Decision Process model in Section 7.3 and 7.4 to accurately depict the movement of the attack within a system.

7.2.1 Social Engineering

Social Engineering refers to the manipulation of people's trust to gain confidential information. A social engineer could have a number of different motivations to attack, such as financial, political, or even plain curiosity/personal interest. Depending on their motivation, the information gained could vary from passwords to sensitive accounts, access to computers to install malicious software or opening doors to stranger's who claim to be someone of authority without proof. Social engineering methods are often used as it is easier to exploit someone's trust than to, for example, hack their computers, [17].

Some of the common Social Engineering attacks include phishing emails which includes receiving 'urgent' emails from strangers claiming they need your help in some way in order to gain access to confidential information such as passwords or bank details. Another type of social engineering attack is baiting, where an attacker leaves a USB stick in a place so that the victim will find it and load it onto their computer to find out what it is. This causes the victim to unintentionally install malicious malware onto their computer. Another example is scareware, where the attacker tricks the victim into believing that his computer has been infected with malware and offers him a way to fix it. However, the victim is simply being tricked into downloading and installing the attacker's malware.

One of the ways to help prevent these types of attacks is through penetration testing carried out by security experts in IT departments. This could allow administrators to understand which type of social engineering techniques are most damaging and which employees tend to be more inclined to fall for them. This will then show them which of the employees could benefit from security awareness training. This will allow them to be more informed about the various types of social engineering attacks possible and avoid falling for them in the future, [57]

7.2.2 Brute Force Attacks

Brute Force attacks, also known as brute force cracking, is used by application programs to decode encrypted data, such as passwords or PINs, by using an exhaustive trial and error method. This could be done by trying every word in a dictionary, or trying commonly used passwords or a combination of letters and numbers, [150]. Although it may be time consuming, it is considered to be an effective cracking method, where hundreds or thousands of password combinations are attempted using automated software. If successful, the hacker will be able to gain access to the victim's personal data and use it to, for example, transfer money to themselves from the hacked account, or to degrade the victim on social media platforms if they are a celebrity, etc.

Brute force attacks may also be used by security analysts to test an organisation's network security.

There are a number of ways to defend against brute force attacks, such as the requirement of complex passwords for sensitive accounts, setting limits to the number of unsuccessful log-in attempts, or locking out those users who exceed the maximum number of attempts to log in to their account, [172].

7.2.3 Man-in-the-Middle Attack

A man-in-the-middle attack, commonly abbreviated to MITM or MIM, is a type of cyberattack where any communication between two systems is intercepted by an outside entity. That is, the attacker is able to hijack, send and receive data meant for someone else without either one of the parties involved being aware of the situation until it is too late, [62]. This could be in the form of emails, social media, web surfing, etc..

Some common types of MITM attacks includes email hijacking, where attackers aim to gain access to email accounts of large organisations, such as banks. They monitor email transactions between organisations and customers until an appropriate opportunity arises for attack. For example, a customer may need to transfer money to a company's account, with which he has been in contact with. As the customer is provided with the bank details, the attacker could re-send and email to the customer telling them to disregard the previous email and provide them with the their own bank details. This was the case in [30], where he had given his solicitor details of his bank account to collect £333,000, unaware that hackers had accessed his email and had been monitoring all his communications, [144]

Wi-Fi Eavesdropping is another form of a MITM attack where hackers set up a fake Wi-Fi connection and wait for users to connect to it. They are then able to steal any personal information.

Other examples include session hijacking, where hackers are able to intercept the user's session with website through, for example, stealing browser cookies, or sniffing, where the attacker can intercept data being sent from or to the user's device by using a software.

Some of the ways to prevent MITM attacks includes using S/MIME to encrypt emails at transit or rest to ensure only the right recipients can read them. This also allows the user to digitally sign their email with a Digital Certificate unique to themselves. Other ways to prevent such attacks include the use of authentication certificates or the use of HTTP Interception through the employment of SSL/TLS Certificates or HSTS to only ensure access to HTTPS.

7.2.4 Denial of Service Attacks

In a Denial-of-Service (DoS) attack, an attacker prevents legitimate users from accessing information or services, such as email, websites or online accounts, by targeting the user's computer and network connection or the sites which they are trying to access. One of the reasons the hacker would do this could be because the hacker has a grievance with the organisation to be attacked, e.g. animal testing or politics, or they may simply find it fun to bring down a service, often to show off their skills to other hackers.

The most common type of DoS attack involves 'flooding' a network with information, or emails with spam messages. This prevents users from accessing a website or receiving legitimate emails by consuming the set quota by the email provider, [122].

These types of attacks can be dated back to 1988, where a graduate student at MIT released a self-reproducing piece of malware which spread itself through the internet and triggered buffer overflows and DoS attacks on affected systems. Although the internet at the time was mainly used by research and academic institutions, it is estimated that up to 10% of the 60,000 systems in the United States were affected, causing damages worth up to $100 million, [107, 163]

Distributed Denial of Service (DDoS) attacks involve the use of a second computer through which an attacker can attack others. For example, by taking advantage of the vulnerabilities and weaknesses of one computer, an attacker could gain access and force the computer to send large amounts of data to websites or send spam to particular email addresses.

One of the ways to prevent such attacks would be through installing anti-virus software and keeping them up-to-date. Firewalls could also be implemented to restrict incoming and outgoing traffic from the internet, [122].

7.2.5 Ransomware Attacks

Ransomware is a type of malware in which the attacker locks the user's keyboard or computer, typically by encryption, and demands a payment to be made before access is given back to the user. These payments are often demanded in virtual currency, such as bitcoin, to keep the cybercriminal's identity safe. Ransomware could infect the user's computer through emails or website or through a backdoor in their computer which the attacker has found, [187].

Cartwright et al. (2018) have recently modelled these types of attacks in a 'kidnapping game' model, by applying and adapting the kidnapping model from Selten (1988).

One of the most recent ransomware attacks includes WannaCry, which infected and encrypted more than a quarter of a million systems globally in May 2017. This included the NHS, which was forced to take services offline during the attack. Again, payments were demanded in bitcoin, of which nearly $100,000 in bitcoin was transferred in the most harmful part of the week and published reports suggest that over $1 billion worth of damages were caused to the thousands of impacted companies, [151].

There are a number of ways in which a user could prevent ransomware attacks to their computer such as regularly backing up their data on external hard drives, the use of layered defence (e..g use of anti-virus, firewalls and web filtering software simultaneously), installing updates from operating systems as soon as they arrive, disconnecting from the network the computer is connected to if attacked, and not clicking on suspicious looking links or unexpected attachments in emails.

7.3 Markov Decision Process

Now that we have provided a brief summary of different types of attacks, outlined the structure of our system and the representation of its different components in Section 3, we are able to move on to develop a model which could describe the movements of attacks in a security system. We have seen from the previous examples of attacks that each attacker has different incentives and would have different views on what it regards as a 'successful attack'. Ransomware attacks, for example, would consider obtaining the requested ransom as a successful attack, whereas brute force attacks would consider the successful decoding of passwords to be job complete. We can use Markov Decision Processes to capture these differences in views of the attackers, where their ideas of 'successful attacks' could be interpreted from the rewards received at different locations.

Markov Decision Processes are classically used to find the optimal action for an agent to take at a location given the transition probability and reward for reaching each of the next locations. We can use this idea to describe the movement of attackers in a system which has been represented in the distributed systems framework.

An outline of the technicalities of a standard MDP is given first to show how the model works in general. We then apply this to our distributed systems model to better understand the movement of the attacker in this framework.

Once we illustrate the integration to the distributed systems framework, we move to explaining the security games and influence models in Sections 7.5 and 7.6, which uses the MDP as a starting point to represent each of the players' movements and will be able to provide us with the expected pay offs of an attacker at each state.

A Markov Decision Process (MDP) is a 5-tuple $(S, A, P(\cdot,\cdot), R(\cdot,\cdot), \gamma)$, where S is a finite set of states, A is a finite set of actions, with A_s being a finite set of actions available from state s. $P_a(s, s') = \mathbb{P}(s_{t+1} = s' | s_t = s, a_t = a)$ is the transition probability of moving from state s to s' by using action a, $R_a(s, s')$ is the immediate reward that is received after the transition from state s to s' and $\gamma \in [0, 1]$ is the discount factor which shows the difference in importance between future rewards and present rewards. The lower the discount factor, the less important the reward is. This is usually the case when working in infinite horizons, where the rewards received in the future are not considered to be as important as the rewards received in the next few time steps.

The goal of this computation is to find a policy, $\pi(s)$, which specifies the optimal action that should be taken in state s. This can be found by finding the actions which maximize the value function.

The reward for a specific action-outcome sequence (path) is constructed from the partial rewards for that path. This allows one to be able to incorporate costs and benefits. For a path of length T, the partial reward is calculated as:

$$r_0^T = \sum_{t=0}^{T} \gamma^t r_t \tag{7.1}$$

From this, we can then move on to calculate the value function (maximum expected reward) for all states $s \in S$ recursively using Bellman's Equation:

$$V_{i+1}(s) = \max_a \Big[\sum_{s' \in S} \mathbb{P}(s'|s,a)(R(s,a) + \gamma V_i(s')) \Big] \tag{7.2}$$

where V_{i+1}^* is the value function for $i+1$ steps to go, V_i^* is the value function for i steps to go and $R(s,a)$ is the expected one-step reward for state s and action a for $R(s,a) = \sum_{s' \in S} \mathbb{P}(s'|s,a) R(s,a,s')$.

Starting from V_0, which represents the rewards for ending in different states, V_{i+1} can be computed for all states until convergence, where $V_{i+1} = V_n$.

The policy can then be calculated by finding the maximizing action for each state s:

$$\pi(s) = \arg\max_{a} \Big[\sum_{s' \in S} \mathbb{P}(s'|s,a)(R(s,a,s') + \gamma V_i^*(s')) \Big] \tag{7.3}$$

The optimal policy π^* is the policy for the optimal value function, V^*, and specifies the action that maximizes the discounted reward over all future states. Now, if the rewards are independent of the the new states, then equations 7.9 and 7.3 can be re-written as

$$V_{i+1}^*(s) = \max_{a \in A} \Big\{ R(s,a) + \gamma \sum_{s' \in S} \mathbb{P}(s'|s,a) V_i^*(s') \Big\} \tag{7.4}$$

and

$$\pi^*(s) = \arg\max_{a \in A} \Big[R(s,a) + \gamma \sum_{s' \in S} \mathbb{P}(s'|s,a) V_i^*(s') \Big] \tag{7.5}$$

7.4 Integration of Markov Decision Processes with Distributed Systems Model

Markov Decision processes can be applied to the distributed systems framework to capture how attackers make decisions at each location given their transition probabilities and rewards. We use this set up as a basis for the Markov Security games, as introduced in Section 7.5, to find the expected pay-off of attackers (or expected loss of defenders) in a zero-sum stochastic game. Details of the implementation of this game will be given in the appropriate sections.

As mentioned previously, a Markov Decision Process is a 5 tuple $(S, A, P(\cdot,\cdot), R(\cdot,\cdot), \gamma)$, which refers to an underlying model of a system. We will relate each of its factors to the components in the distributed systems model, as introduced in Section 3, to show the reader how these two models are integrated together.

Suppose we have a system with 3 different locations (containing resources that could be compromised) with connections as shown in Figure 7.1. For a given attacker in the system, the states are given as $s_1, s_2, ..., s_{N_S}$ where ($N_S = 2^m$ and m =number of locations) and $s_k \in S \in \{0,1\}$ with $k = 1, ..., N_S$. We let 1 indicate that a location has been compromised and 0 not compromise.

The different possibilities of interactions between states is shown in Figure 7.2. Here, the states are given in terms of the situation of each of the locations, for example, $S_6(1,0,1)$ shows that Location 1 and 3 have been compromised and Location 2 remains not compromised. Here, the state can either remain in the same state or change into state $S_8(1,1,1)$, where all three of the locations have been compromised.

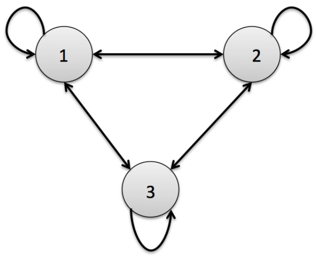


Figure 7.1: An example system with 3 different interconnected locations

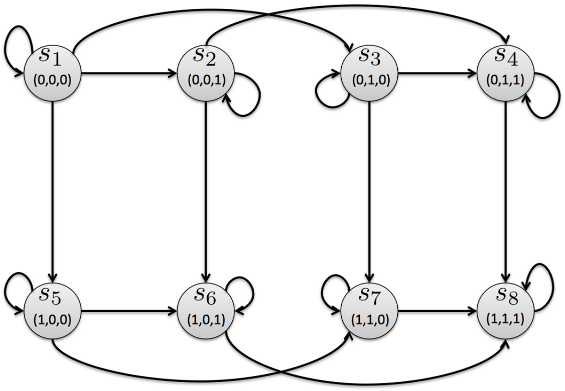


Figure 7.2: Different interactions between states for Figure 7.1

We take our attacker to have 2 different actions; *Attack* or *Not Attack*. The attacker can decide to use a vulnerability to advance through the system towards their target location or it can decide to take no action. Throughout this work, we assume that the attacker can only use one single attack at a time.

We let the transition probability be the probability that an attacker moves from state s_i to s_j given that he has used action 'attack', where $i, j \in \{1, ..., N_S\}$. This can be formally represented as:

$$\mathbb{P}(s_i, s_j) = [s_{t+1} = s_j | s_t = s_i, a_t = Attack].$$

Note that, a successful transition between locations implies the damages caused to resources at those locations (i.e., the resources are modelled implicitly in this representation. These will be captured explicitly when expressed in terms of the damages caused to the security attributes in Section 4).

$R(s_i, s_j)$ represents the damage caused to the attacked location when moving from state s_i to s_j. For example, if the attacker attacks location 2 from location 1, he would receive the reward of moving from state s_5 to s_7.

Finally, the discount factor, γ, will be set according to the attacker's preference in causing immediate or future damage to the system.

MATLAB is used to find the optimal policy for the optimal value function of a given system.

7.5 Markov Security Games

Having set up the basic theoretical foundation for the development and analysis of player strategies, we can naturally extend this model to a multi-agent Markov Security game to find the optimal defence and attack strategies for the defender and attacker, respectively, at each state. The expected pay-off of the attacker/expected loss of the defender at each stage can also be calculated to be used later in the economic security model in Section 4.

In what follows, we provide a review of some of the main concepts in Markov security games. This is then extended by incorporating the influences between security assets and vulnerabilities at different locations in a system in Section 7.6 to better describe the interdependencies within organisations as discussed in Section 2.

Markov Security Games, or Stochastic Security Games, have a rich background in game theory, being first introduced by Shapley (1953).

Similar to MDPs, a stochastic game is a tuple $(n, S, A_{1,\dots,n}, T, R_{1,\dots,n}, \gamma)$, where n is the number of agents, S is a set of states, A_i is the set of actions available to agent i with A being the joint action space, $A_1 \times \cdots \times A_n$, T is a transition function $S \times A \times S \rightarrow [0, 1]$, R_i is a reward function for the ith agent $S \times A \rightarrow \mathbb{R}$ and γ is the discount factor, [33].

In a stochastic game, multiple agents select their own actions which determine the next state and reward. The goal for each agent is to select actions to maximise their discounted future reward with discount factor γ. These games can be simplified back to a Markov Decision Process if all, except one, of the players adopt a fixed strategy, turning it into an optimisation problem for the remaining player, [9].

Fixed strategies could, however, be exploited by other agents in multi-agent settings. For example, let us consider a simple 'Rock-Paper-Scissors' game. If one of the players was to always choose the 'Rock' outcome, it would allow the other players to use this information for their own benefit and choose their outcomes accordingly.

Therefore, mixed strategies or policies are commonly used to obtain optimal payoffs. We represent a mixed policy, ρ, as $\rho : S \rightarrow PD(A_i)$, where PD is a probability distribution, to show that it is taken to be a function that maps states to mixed strategies which are probability distributions over the players' actions.

Mixed strategies cannot produce optimal strategies that are independent of the other players' strategies. Nevertheless, a notion of best response can be defined if a player's strategy is optimal given the other players' strategies. This notion is commonly known as a Nash equilibrium (Nash,1950), which has driven much of the development of matrix games, game theory and stochastic games, as previously discussed in Section 2.

For the purposes of our model, we consider a two-player, zero-sum stochastic game, where one of the players is the attacker and the other the defender. As mentioned previously, in this model, the gain of one player (attacker) is equal to the loss of the other player (defender). We have chosen to carry out a zero-sum game to capture the damages the attacker causes as it moves between locations. We understand that one of the main limitations of this game framework is its inability to capture the win-win or lose-lose outcome. However, as an initial model, we believe the use of a zero-sum game is a good starting point to record the level of damage caused by the attacker, which will then allow us to see the optimal way in which an organisation could recover itself using Impulse-Response functions.

Further details of the set up of this game, applied to the security network, are given in the next section.

7.5.1 The Markov Security Game Model

This section sets up a Markov Security Game model for a general security network. Details on how to solve these types of games is provided in Section 7.5.2.

We consider a stochastic game played between an attacker, P^A, and defender P^D, on a finite state space, $S = \{s_1, s_2, ..., s_{N_S}\}$, where N_S is the total number of states. Similar to the MDP in Section 7.4, each of the states represents an operational mode of the security system network showing which locations have been compromised, 1, or not compromised, 0. We assume that the states evolve according to a discrete-time Markov chain. This allows for the utilization of well-established analytical tools to study the problem.

Each player has a finite number of actions to choose from. We set the action space of the attacker to be $A^A := \{a_1, ..., a_{N_A}\}$, where each of its elements represent different types of possible attack, and N_A is the total number of possible attacks.

Similarly, the action space of the defender is defined as $A^D := \{d_1, ..., d_{N_D}\}$, where each of the elements show the different types of defensive measures that can be used and N_D is the total number of defensive measures possible.

We can then use the mapping $\mathscr{M} : S \times A^A \times A^D \to S$ to show the evolvement of the state space as a result of player's actions.

We define a probability distribution, $p^S := \{p_1, ..., p_{N_p}\}$ on the state space S, with $0 \leq p_i^S \leq 1 \; \forall i$ and $\sum_i p_i^S = 1$. The mapping, $\mathscr{M}$, can then be represented by an $N_S \times N_S$ (Markov) transition matrix

$$M(a, d) = [M_{i,j}(a, d)]_{N_S \times N_s}$$

where $a \in A^A$ and $d \in A^D$.

The next state probability vector can then be defined as $p^S(t+1) = M(a, d)p^S(t)$, where $t \geq 0$.

We define the gain of the attacker/loss of the defender as a zero-sum game matrix, $G(s)$, which is given as:

$$G(s(t)) = [G_{a,d}(s(t))]_{N_A \times N_D} \tag{7.6}$$

where $s \in S, a \in A^A, d \in A^D$ and $N_A \times N_D$ is the dimension of the game matrix. For example, if P^A chooses action a_3 and P^D chooses the action d_5, when in state s_2, then the outcome of the game is $G_{3,5}(2)$.

We take this stochastic game to consists of game elements, as in [9, 137], which are counterparts of the zero-sum matrix $G(s)$.

We denote the probability of playing the j-th element when currently in element i under the given actions of the players as:

$$q_{ij}(a, d), \quad a \in A^A, d \in A^D, i, j, \in S$$

where $\exists \; q_{i,0}(a, d) > 0$ to a state 0 at which the game terminates regardless of the actions of the players. We introduce this non-zero termination probability to ensure that the probability of infinite play is zero and all expected costs are finite even without a discount factor. These state transitions approximately correspond to the elements in $M(a, d)$.

We assume that the strategies of each of the players are state dependent and set them as probability distributions defined on their respective action sets. That is, the strategy of P^A is set to be

$$p^A(s) := [p_1^A(s), ..., p_{N_A}^A(s)]$$

and the strategy of P^D is given by

$$p^D(s) := [p_1^D(s), ..., p_{N_D}^D(s)]$$

for $0 \leq p_i^A, p_i^D \leq 1 \forall i$ and $\sum_i p_i^A = \sum_i p_i^d = 1$.

7.5.2 Solving Markov Security Game Model

We can now solve the zero-sum Markov game that has been set up in the previous section. We give a brief overview of the solution method for solving Markov Security games and present the Value Iteration Algorithm (Algorithm 1), as in [9, 137], and apply it to our model to provide the reader with a better understanding of the solution. We implement this algorithm in MATLAB to be used later in Section 7.7.

The solution methods, such as value iterations, for Markov games are closely related to the solution methods for MDPs, with the requirement of slight modifications.

We assume that the the defender in the zero-sum Markov game aims to minimize its own aggregate cost, Q, whilst facing the attacker, who tries to maximise it. Due to the zero-sum nature of the game (loss of P^D = gain of P^A), it will be sufficient to describe the solution algorithm for one player only. Therefore, we only consider the analysis of the solution algorithm for the defender only.

We begin by defining the aggregate cost for the defender, which is similar to the one in Markov Decision Processes. We assume that the game is played over an infinite, discrete-time horizon whose aggregate cost at the end of the game is given by the sum of all realized stage costs multiplied by the discount factor, $\gamma \in [0,1]$:

$$Q := \sum_{t=1}^{\infty} \gamma^t G_{a(t),d(t)}(s(t)), \quad a(t) \in A^A, d(t) \in A^D, s(t) \in S \tag{7.7}$$

where $G_{a(t),d(t)}$is the game matrix as define in equation (7.6).

Theoretically, the defender can choose different defence strategies at each state in order to minimize its final cost Q. However, it has been shown that a stationary strategy, $p^A(s) = p^A(s(t)) \forall t$, is optimal. Therefore, there is no need to compute a separate optimal strategy for each stage. The stationary optimal strategy can be obtained recursively using dynamic programming where a zero-sum matrix game is solved at each stage. This means that the optimal strategy can therefore be mixed, unlike an MDP.

We can then calculate the optimal cost, $Q_t(a,d,s)$, at a given time, t, using dynamic programming recursion:

$$Q_{t+1}(a,d,s) = G_{a,d}(s) + \gamma \sum_{s' \in S} M_{s,s'}(a,d) \times \min_{p^D(s')} \max_{a} \sum_{d \in A^D} Q_t(a,d,s') p_d^D(s'). \tag{7.8}$$

for $t = 0, 1, ...$and a given initial condition, Q_0. Note that the Q values are now also defined over player actions in addition to the states.

This recursion can be split into two parts to represent the counterparts of the Bellman's equation for the Markov game:

$$V(s) = \min_{p^D(s)} \max_{a} \sum_{d \in A^D} Q_t(a,d,s) p_d^D(s) \tag{7.9}$$

where

$$Q_t(a,d,s) = G_{a,d}(s) + \sum_{s' \in S} q_{s,s'}(a,d) V(s'), \quad t = 1, 2, ... \tag{7.10}$$

The convergence points of equations (7.9) and (7.10) give the optimal minimax solution for the defender since

$$\sum_{j \in S} q_{ij}(a,d) < 1.$$

By swapping the positions of min and max in equation (7.9), we can find the corresponding mixed strategy of the attacker. That is, we set the maximization to be over $p^A(s)$ and the minimization over d. However, the values of V^* and Q^* do not change due to the two-player, zero-sum nature of the game.

The Bellman equation can be solved using the value iteration algorithm as given by Alpcan and Başar (2011), and can be found in Algorithm 1.

Algorithm 1 Value Iteration Algorithm

```
Given arbitrary $Q_0(a,d,s)$ and $V(s)$
repeat
    for $a \in A^A$ and $d \in A^D$ do
        Update $V$ and $Q$ according to (7.9) and (7.10)
until $V(s) \rightarrow V*$, i.e. $V(s)$ converges.
```

7.6 Modelling Interdependencies in a Security Network Using Linear Influence Models

In many cases, the level of security placed on one type of resource at a location would depend on the security measures placed on resources at locations directly linked to it to a certain extent.

Let us consider, as an example, the use of the same username and/or password for multiple websites. If an attacker is aware of this, and is able to gain access to a user's password through a low security website, such as a high school reunion website, he may also be able to gain access to the victim's username and/or password to more sensitive websites, such as a bank or online retail organisation.

Another example would be the protection of valuables, such as expensive jewellery or cash, which have been locked inside a high-security vault and placed inside an access controlled room. In such a situation, the level of protection placed on the valuables is partially dependent on the security measures placed on both the vault and room door. If an attacker wants to gain access to the valuables, he would have to bypass two different types of defences.

We can see that if an attacker is able to get through one type of defence, such as the access controlled door, the overall level of protection of the valuables immediately decreases, as the attacker now only has one more type of defence to overcome in order to get hold of the valuables.

We can capture these types of interdependences between locations using linear influence networks as in [9, 10, 22, 24]. Although there may exist a non-linear relationship in some cases, it seems reasonable to use a linear approximation within certain decision-making ranges. For example, Bambos et al. (2008) show how one organisation's investments are amplified by some linear function of its neighbour's investments when studying the risk in computer security settings. That is, the investments made to protect the resources at one location had a linear effect on the investments of the protection of resources at the location directly linked to it.

The next few sections give an overview of the way these linear influence model have generally been set up, in previously mentioned papers, to describe the relationships between security assets and vulnerabilities in networks. We then apply this model to a security network, in the distributed systems framework, to gain a better understanding of the interdependencies between security controls and their effects on the resources they are protecting in case of an attack. We then use this in the two-player stochastic zero-sum game as set up in Section 7.5 to calculate the expected payoff for attackers in interdependent systems.

7.6.1 Security Assets

We begin by considering the relationships between security assets in systems. Here, the term 'security assets' is used in a broad sense in that it refers to a vertex in an interconnected system which plays a security related role, such as security controls at different locations. We model the network of resources as a weighted directed graph, $\mathscr{G}_s = \{\mathscr{N}, \mathscr{E}_s\}$, where $\mathscr{N}$ is the set of vertices with cardinality n and the set of edges, $\mathscr{E}_s$, represents the interdependencies between vertices. The weight of each edge, $e_{ij} \in \mathscr{E}_s$, is denoted by a scalar w_{ij} that signifies the influence of vertex i on vertex j, where $i, j \in \mathscr{N}$. That is, the level of protection placed on the resources at location j as a result of the defences placed at location i, given that there is an edge between the two locations.

We understand that the level of protection placed on resources from one location to another is not consistent throughout a given organisation. For example, it may be easier for attackers to move between locations in an organisation once they have successfully bypassed the security controls on the outer edges (i.e. to enter the organisation). This variability in the level of influence, or protection, across locations in an organisation is captured in the effective security assets vector, defined below, as the product of the influence matrix and independent security assets vector.

We define the entries of the influence matrix, W_{ij}, as follows:

$$W_{ij} = \begin{cases} w_{ij} & \text{if} \quad e_{ij} \in E_s \\ 0 & \text{otherwise,} \end{cases} \tag{7.11}$$

where $0 < w_{ij} \le 1 \; \forall i, j \in \mathscr{N}$ and $\sum_{i=1}^{n} w_{ij} = 1, \forall j \in \mathscr{N}$. The entry $w_{jj} = 1 - \sum_{i=1, i \neq j}^{n} w_{ij}$ is taken to be the self-influence of a vertex on itself (i.e., the level of protection of the resources at location j due its own defences).

Let the vector $x = [x_1, x_2, ..., x_n]$ quantify the value of the *independent* security assets of the system (i.e. the value of the individual vertices in $\mathscr{N}$). The effective security assets, $y = [y_1, y_2, ..., y_n]$, are then defined as:

$$y = Wx \tag{7.12}$$

as in [9, 10].

We can then prove that $\sum_{i=1}^{n} y_i = \sum_{j=1}^{n} x_j$ with the condition $\sum_{i=1}^{n} w_{ij} = 1, \forall j \in$

$\mathscr{N}$ as follows:

$$\sum_{i=1}^{n} y_i = \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}x_j = \sum_{j=1}^{n}\sum_{i=1}^{n} w_{ij}x_j$$
$$= \sum_{j=1}^{n} x_j \sum_{i=1}^{n} w_{ij} = \sum_{j=1}^{n} x_j.$$

i.e. the sum of all the effective security assets in a system is equal to the sum of all the independent security assets in the same system.

If a vertex is compromised, we remove the affected vertex and all the edges connected from the graph. Hence, the security loss of the network will be the vertex's effective security asset value instead of its independent one. Conversely, if a vertex is secured, it regains its original influence on other vertices. In either case, the entries of the influence matrix, W, are normalized to keep it stochastic [9].

For example, let us consider a GSM network, where a number of base transceiver stations are controlled through a base station controller. If the controller fails, all the other stations connected to it will be out of service. In contrast, if a single transceiver is compromised, it should not affect the communications between the users under other transceivers.

7.6.2 Linear Influence Model Example for Security Assets in Distributed Systems Framework

An example of a network with three locations and its associated computations will be given in this section to provide a better understanding of the introduced linear influence model.

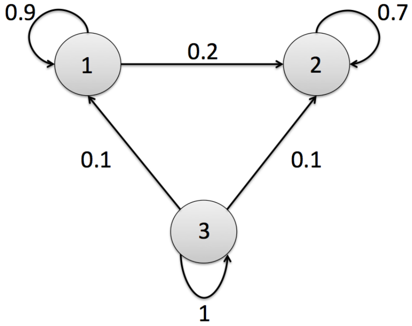


Figure 7.3: Linear influence network example for security assets where each of the connections describe levels of influences between locations

We use the example from Alpcan et al. (2009) to describe our network in Figure 7.3, where each of the connections between locations describe the levels of influence between them. Each of the security assets are given in terms of their location and stage of the game, for example, $y_2^{(1)}$ shows the effective security asset value at location 2 during the first stage of the game.

The influence equation (7.12) then becomes:

$$\begin{pmatrix} y_1^{(1)} \\ y_2^{(1)} \\ y_3^{(1)} \end{pmatrix} = \begin{pmatrix} 0.9 & 0.2 & 0 \\ 0 & 0.7 & 0 \\ 0.1 & 0.1 & 1 \end{pmatrix} \begin{pmatrix} x_1^{(1)} \\ x_2^{(1)} \\ x_3^{(1)} \end{pmatrix} \tag{7.13}$$

Suppose that Location 1 is attacked. This results in the removal of the vertex representing Location 1, along with its connections to the other vertices, from the graph. The influences of each of the remaining locations on each other will now have to be normalised to continue to satisfy the condition $\sum_i w_{ij} = 1$. Therefore we now have

$$w_{22} = \frac{w_{22}}{w_{22} + w_{32}} = \frac{7}{8}$$

and

$$w_{32} = \frac{w_{32}}{w_{22} + w_{32}} = \frac{1}{8}.$$

This new network with the two remaining locations and their influences is illustrated in Figure 7.4.

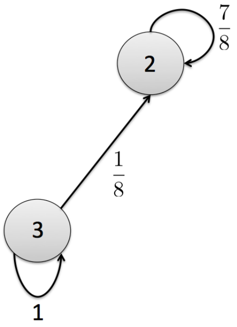


Figure 7.4: Linear influence network example for security assets after Location 1 has been attacked.

Taking this attack into consideration, the new influence equation now becomes:

$$\begin{pmatrix} y_2^{(2)} \\ y_3^{(2)} \end{pmatrix} = \begin{pmatrix} \frac{7}{8} & 0 \\ \frac{1}{8} & 1 \end{pmatrix} \begin{pmatrix} x_2^{(2)} \\ x_3^{(2)} \end{pmatrix} \tag{7.14}$$

The attack on Location 1 would mean that the *independent* security asset of Location 3 would remain the same as the security asset at Location 1 did not have any effect on Location 3, so $x_3^{(1)} = x_3^{(2)}$.

However, the *independent* security asset of Location 2 will now have decreased by an amount corresponding to the influence of Location 1 on Location 2. That is,

$$x_2^{(2)} = x_2^{(1)} - 0.2x_2^{(1)} = 0.8x_2^{(1)}.$$

This results in the *effective* security asset of Location 2 to stay the same since we have from equation (7.14) that

$$y_2^{(2)} = \frac{7}{8}x_2^{(2)} = \frac{7}{8} \times 0.8x_2^{(1)} = 0.7x_2^{(1)}.$$

The *effective* security asset of Location 3 decreases by an amount corresponding to its influence on Location 1:

$$y_3^{(2)} = \frac{1}{8}x_2^{(2)} + x_3^{(2)} = \frac{1}{8} \times 0.8x_2^{(1)} + x_3^{(1)} = 0.1x_2^{(1)} + x_3^{(1)}.$$

Let us now consider an attack on Location 3. The corresponding vertex and connections with Location 2 are removed from the graph and result in a single vertex remaining (Location 2) with a self influence of 1, that is, $w_{22} = 1$. The independent and effective security asset of Location 2 can then be calculated to be:

$$x_2^{(3)} = x_2^{(2)} - \frac{x_2^{(2)}}{8} = \frac{7}{8}x_2^{(2)} = 0.7x_2^{(1)},$$

and

$$y_2^{(3)} = x_2^{(2)}.$$

7.6.3 Vulnerabilities

Now that we have set up a model to represent the security assets in a network, we can set up a similar model to capture the interdependencies between vulnerabilities at different locations in a system. For example, let us consider a connected computer that is compromised, the data stored in this computer can be used in attacks against other computers. Therefore, the latter computers will become more vulnerable to attacks. This dependency has previously been modelled using linear influence models in [9, 10, 22].

Let us consider a weighted, directed graph $\mathscr{G}_v = \{\mathscr{N}, \mathscr{E}_v\}$, as before, where the edges, $e_{ij} \in \mathscr{E}_v$, now denote the amount of influence between the vertices, v, in terms of their levels of vulnerabilities. That is, the effect on the vulnerability of the resources at location i due to a successful attack on the resources at location j, given that there exists an edge between the two locations.

The vulnerability matrix is then defined as:

$$V = \begin{cases} v_{ij}, & \text{if} \quad e_{ij} \in \mathscr{E}_v \\ 1, & \text{if} \quad e_{ii} \\ 0, & \text{otherwise} \end{cases} \tag{7.15}$$

where $0 \le v_{ij} \le 1$ represents the vulnerability of vertex i due to vertex j as a result of the interdependencies in the system. The self-influence is set to be one, $v_{ii} = 1, \forall i$, which can be seen to be the default level of vulnerability of a vertex independent of others. The aggregate influence on vertex i from all other vertices is defined as $v_i = \sum_{j=1}^{n} v_{ij}$.

Note that, since $v_i \ge 1$ as $v_{ii} = 1 \forall i$, this vulnerability matrix is not stochastic.

This model shows that the more connected a vertex is, the more vulnerabilities it has due to outside influences. For example, the more connected a computer is with the outside world, the higher its risk of getting attacked.

In the next section, we set up a zero-sum stochastic security game and integrate the influence models with it to obtain a value for the pay off of an attacker at a location with certain defensive actions. Note that, the underlying framework for this will still be in terms of the distributed systems, where the defenders are protecting resources at different locations whilst the attackers are trying to attack it.

This model can be used in the enriched Economic Security Model in Section 6 to get a more accurate estimation of the attacker's pay offs in the presence of defences. A better explanation of the integration of these two models is given in Section 8.

7.7 Integration of Markov Security Game and Linear Influence Model

The integration of the Markov Security Game and Linear Influence model will allow us to find the damages caused by an attacker in an organisation and the way it could move once it has gained entry.

This can then be used in the final integration to obtain the entries for the placeholder matrix in the enriched economic security model, which will be discussed in more detail in Section 8.

The validity of this model could then be determined using a number of validation techniques, such as stability conditions, sensitivity analysis and thought experiments as in Sections 4.9 - 4.11 and 6.0.4, to ensure that the values that have been obtained are accurate.

We begin by setting up a zero-sum stochastic game for the security problem similar to [153] where the existence, uniqueness and structure of the solution for this game have been proven in [10].

We consider a security network in the distributed systems framework as set up in Section 3, with n locations which contain resources to be protected.

We construct a two-player zero-sum game between an attacker, P^A, and a defender, P^D, where they each have $n+1$ actions; the attacker can either choose to attack location i with action $a_i \in A^A$, or do nothing, $\emptyset$. Similarly, the defender can choose to either defend location i with defensive action $d_i \in A^D$ or do nothing, $\emptyset$. We assume that the attacker will have no motivation to attack a location if it has already been compromised and that he is only able to attack one location at a time.

If the attacker is unable to successfully attack a location, there is a probability of $p_r^{(k)} \in (0,1)$ that the attacker will remain at location k. This means that the defender has defended the location to be attacked successfully, causing the attacker to remain at its original location. The defender can also restore all the compromised locations with probability $p_r^{(1)} \in (0,1)$. If, in addition, he is also able to stop the attacker from causing any more damage to the network, the game will end with probability $p_e \in (0,1)$.

We further define the probability that a location gets compromised if defended as p_d^k and not defended as p_{nd}^k, where k represents their current locations.

The instant pay off the attacker receives can then be defined as:

$$G_{a,d}(s) = c_i(a,d,s)y(s) \tag{7.16}$$

where $a \in A^A$, $d \in A^D$, $s \in S$ and $c_i(a_i, d_i, s) = p_*^s v_i$ is the probability that location i is compromised given the actions of the attacker and defender at state s, with $*$ being any of $\{d, nd\}$. Finally, $y(s)$ is the effective security asset as defined in equation (7.12).

A numerical example is given in the following section to illustrate how this integrated security game could be solved.

7.7.1 Numerical Example

We consider a network with three locations as in Figure 7.2 with the system states given as

$$S = \{000, 001, 010, ..., 111\}$$

where 1 shows that a location has been compromised and 0 that it has not. We enumerate the states in the order given such that 000 is state 1, 001 is state 2, and $010, 100, 011, 101, 110, 111$ represent states 3, 4, 5, 6, 7 and 8, respectively.

We set the influence, matrix which quantifies the redistribution of the security asset values among different locations, as defined in Section 7.6.1, to be:

$$W = \begin{pmatrix} 0.8 & 0.2 & 0 \\ 0 & 0.6 & 0 \\ 0.2 & 0.2 & 1 \end{pmatrix}$$

with independent security asset values for each of the locations as $x = [8, 8, 15]$.

The vulnerability matrix as defined in Section 7.6.3 is set to be:

$$V = \begin{pmatrix} 1 & 0.1 & 0 \\ 0.4 & 1 & 0 \\ 0.1 & 0.3 & 1 \end{pmatrix}$$

We choose the probabilities to be as follows:

- Probability that a location k is successfully compromised, given that it is defended is $p_d^k = 0.2$.
- Probability that a location k is successfully compromised, given that it was not defended is $p_{nd}^k = 0.4$.
- Probability that the defender successfully defends a location and restores the system to state 1, (000), is $p_r^1 = 0.7$.
- Probability that the defender successfully defends and causes the attacker to remain at its current location, k, is $p_r^k = 0.2$.
- Probability that the game ends is $p_e = 0.3$.

As mentioned in the previous section, we let the attacker have actions $A^A = \{a_1, a_2, a_3, \emptyset\}$, which means that it can attack locations 1, 2, 3 or do nothing, respectively. Similarly, we let the defender have the following actions $A^D = \{d_1, d_2, d_3, \emptyset\}$

which means that they can defend either of locations 1, 2, 3 or do nothing.

Scenario 1

Let us consider an attack on location 1, given that the defender defends it.

The possible states are $\{s_1 = 000, s_4 = 100\}$. We can compute the instant pay off for the attacker as:

$$G_{11}(1) = c_1(1,1,1)y(1)$$

where $c_1(1,1,1) = p_d^1 v_1$. That is, the probability that location 1 is compromised at state 1, given that the attacker attacks it, and the defender defends it multiplied by the aggregate influence on the vulnerability of location 1 from other connected locations.

The probability of remaining at state 1 given that the attacker is attacking location 1 and the defender is defending it can be calculated as follows:

$$q_{11}(1,1) = (1 - c_1(1,1,1))(1 - p_e)$$

That is, the probability of location 1 not compromised multiplied by the probability that the game does not end.

The probability of moving from state 1 to 4, given that the attacker is attacking location 1 and the defender is defending it, is calculated as:

$$q_{14}(1,1) = (c_1(1,1,1))$$

That is, it is equal to the probability of compromise of location 1 given that the attacker is attacking it and defender is defending it at state 1.

Finally, the probability of moving from state 1 to any other state, given that the attacker is attacking location 1 and the defender is defending it is zero:

$$q_{1j}(1,1) = 0 \quad \forall j \neq 1,4$$

However, if the defender had chosen to defend, for example, location 2 instead, whilst location 1 is being attacked, we would have the following results:

$$\begin{aligned} G_{12}(1) &= c_1(1,2,1)y(1), \\ q_{11}(1,2) &= (1 - c_1(1,2,1))(1 - p_e), \\ q_{14}(1,2) &= c_1(1,2,1), \\ q_{1j}1,2 &= 0 \quad \forall j \neq 1,4 \end{aligned}$$

Scenario 2

Let us now suppose that the attacker has successfully attacked location 1 so that the system is now in state 4. The next state could be any of $\{000, 100, 101, 110\}$, represented by state 1, 4, 6, 7, respectively. In order to reach those states, the attacker could either attack location 2, 3 or do nothing and the defender could either defend location 2, 3 or do nothing.

Let us consider the case where the attacker attacks location 2 and the defender defends it, we then have the following outcomes and probabilities:

$$\begin{aligned}
G_{12}(1) &= c_2(2,2,4)y(4), \\
q_{47}(2,2) &= c_2(2,2,4), \\
q_{41}(2,2) &= (1 - c_2(2,2,4))p_r^4, \\
q_{44}(2,2) &= (1 - c_2(2,2,4))(1 - p_r^4 - p_e^4), \\
q_{4j}(2,2) &= 0 \quad \forall j \neq 1,4,7
\end{aligned}$$

where $c_2(2,2,4) = p_d^4 v_2$, i.e., the probability that location 2 is compromised at state 4, given that it was being defended.

Similar calculations can be made to obtain the different possible outcomes. We can then solve this stochastic security game using Algorithm 1 presented in Section 7.5.2 to obtain the optimal strategies for both the players and the payoffs of the attacker/expected losses of the defender at each state.

8 Conclusions and Recommendations

The aim of this thesis was to provide an integration of different concepts taken from three separate areas of research; Distributed systems, Stochastic Security Games and the Economics of Information Security to help in understanding the following points:

1. How an organisation's security network can be represented in the distributed systems framework and used as an underlying network.
2. The behaviour of organisation with different preferences when attacked and the investments they are required to make to overcome these.
3. The movement of attackers in a secure system and their interaction with various types of defences placed at different locations.
4. The expected damages caused by an attacker in a system with interdependent defensive measures placed at different locations.

We started to fulfil these aims by using the distributed systems framework, as describe in Section 3, to represent the network in terms of its locations, resources, processes and environment. We applied this framework to a number of real life networks to show its applicability (Point 1).

We then went on to explain the set up of our economic model, which captures the effects of a dynamic, stochastic shock on organisations with different preferences in their security attributes in Section 4 (Point 2). A sensitivity analysis, stability analysis and a set of thought experiments have been carried out to assess the model's validity and robustness in various settings. This model is then enriched with the integration of the distributed systems framework in Section 6 to allow for the organisation's network structure to be captured. We have discussed the changes that this enrichment brings to the initial model and applied them accordingly to ensure we maintained the validity and reliability of the model in Section 6.0.4.

Different types of attacks were then introduced and a suggestion was made to model the movement of attackers in a system and the damages it could cause in Section 7. We used the integration between the distributed systems model and attack model to have a way of describing the possible movement of an attacker in a system (Point 3).

The notion of linear influence models has also been introduced, and integrated with the distributed systems framework, to obtain the expected pay-off of attackers in a system in Sections 7.6 – 7.7 (Point 4). This can be used in future work to describe the damages caused by an attacker at each location, in the presence of defences, as part of a zero-sum stochastic security game.

Currently, the model does not incorporate the possibility of multiple attackers at any given point. However, this could be a point of development in future research, where the placeholder matrix, $R_{\star ij}$ as described in Section 6, could be transformed into a 3D matrix as an initial starting point. Here, each layer of the matrix could represent one type of attack and its movement within a system at a given time, and the damages it causes in the presence of defences, as part of a zero-sum stochastic security game. Inevitably, the model would then become much more complex, however, it would allow one to capture various evolutions at once, which could give a more realistic set of results for the defender's actions at each location over time.

The main recommendations for an extension to this work would be the integration of all three models discussed previously; Distributed Systems, Stochastic Security Games and the Economics of Information Security. This fully integrated model would allow to find the following:

1. Observe the damages caused at *each location* of a system over time given the defensive measures present.

2. Analyse the path taken by the attacker to get to its target location and the damages caused at each location whilst reaching its target.

3. Understand how an organisation recovers a compromised location and the time taken to do this in the form of impulse-response functions.

4. Understand the level of investments to be made in defensive measures at each location to prevent such attacks in the future.

The organisations considered would be conforming to the assumptions set out in Section 4, with differing preferences for each of the security attributes. The magnitude of the damage caused by the attacker at each location can then be obtained by calculating the pay-off of attackers in networks as described in the attacker model in Section 7.

One way of implementing this integration would be by representing an organisation's security system in the distributed systems framework as explained in Section 3. Here, the organisation can be taken to be the *environment* from which services are delivered to and from and its assets to be protected to be the *resources*, such as computers, sensitive data, websites, etc. These would be kept at different *locations* within the organisation and the *processes* could be taken to be any actions that manipulate the resources, such as printers copying documents or the insertion of keys in locks.

Next, the system's network can be represented as a directed graph, where each of the vertices represent different locations, that contain resources, within the organisation (e.g. different rooms containing employees' personal belongings, or desktop computers containing sensitive data) and each of the directed edges show any links between these (e.g. corridors between two rooms or wireless connections between two computers), as explained in Section 3.2.2.

Each of the locations in the system can be represented by the elements of the C, I, A, $\dot{K}$ matrices as introduced in Section 6, where C_{ij} represents the level of protection of the confidentiality of the resources at location i from attacks from location j. Similar definitions are set for the integrity and availability of the system, and K_{ij} represents the amount of investments made to protect the resources at location i from attacks from location j.

Note that the resources are no longer explicitly defined, rather they are now captured through the representation of their properties to be protected in terms of the three attributes at each location.

Next, a placeholder, $R_{\star}$, where $\star = \{C_{ij}, I_{ij}, A_{ij}, \dot{K}_{ij}\}$, can be used to represent the presence of an attack on any of $C, I, A, \dot{K}$, respectively.

The expected damages caused by attackers at each location can then be found by using the instant pay-off of attackers computed in Section 7.7. That is, the damages caused to each location by the attacker given the defensive measures present at that location, in terms of the confidentiality, integrity, availability and change in investment.

In order to find the instant pay-off, some of the the concepts introduced in Section 7.5.1 to describe the interaction between an attacker and defender at different locations in the system can be used.

In particular, the concepts of the influence matrix, W, independent security asset, x, and the vulnerability matrix, V, as defined in Section 7.6, can be used to describe the interdependencies between locations in terms of their levels of protection of the confidentiality, integrity, availability and investment at each location.

Let $\mathscr{G} = \{\mathscr{N}, \mathscr{E}\}$ be a directed graph representing a system network, with system states $S = \{s_1, .., s_{N_S}\}$, as defined in Section 7, and let $\star$ be any of $\{C, I, A, \dot{K}\}$.

We take $W_{\star_{ij}}$ to be the influence matrix, where $w_{\star_{ij}} \in W_{\star_{ij}}$, describes the level of influence of the protection placed on the resources at location j on the level of protection placed on the resources at location i, given that there is link (edge) between them.

That is,

$$W_{\star_{ij}} = \begin{cases} w_{\star_{ij}} & \text{if} \quad e_{ij} \in \mathscr{E} \\ 0 & \text{otherwise,} \end{cases} \tag{8.1}$$

The independent security asset value, $x_{\star_i}$, at each location is taken to be the standalone level of protection of the resources at that location in the system. These are also split up in terms of the security attributes and investment to better understand the effect of each of them on the system.

We use the influence matrix and vector of independent security asset values to find the effective security asset values, $y_{\star_i}$, in terms of each the security attributes and investment. That is, the redistributed effect of protections placed at each location on other locations in the system as a result of the interdependencies within that system.

We define the vulnerability level, $v_{\star_{ij}} \in V_\star$, to be the vulnerability level of resources at location i if the resources at location j have been compromised (i.e., how vulnerable location i is to attack if location j has successfully been attacked).

Our vulnerability matrix, in terms of the security attributes and investment, then becomes:

$$V_{\star_{ij}} = \begin{cases} v_{\star_{ij}}, & \text{if} \quad e_{ij} \in \mathscr{E}_v \\ 1, & \text{if} \quad e_{ii} \\ 0, & \text{otherwise} \end{cases} \tag{8.2}$$

The aggregate influence on location i from all other connected locations is then defined as

$$v_{\star_i} = \sum_{j=1}^{m} v_{\star_{ij}},$$

where m is the number of locations in the system.

We then let the probability of attack on the confidentiality, integrity, availability or level of investments made at location i be

$$c_{\star_i}(a_i, d_i, s_i) = p_*^s v_i,$$

where $*$ is any of $\{d, nd\}$ (i.e. defender chooses to defend or not defend location i when it is being attacked).

The instant pay-off the attacker receives with respect to each of the security attributes or investment can then be given by

$$G_{\star_{a,d}}(s_i) = c_{\star_i}(a, d, s)\, y_\star(s). \tag{8.3}$$

That is, the probability that a location will get compromised multiplied by the value of the effective security asset at that location. Further probabilities of various possible scenarios can also be computed, following the example calculations in Section 7.7.

By analysing the movement between states we can find which location has been damaged from where and the payoffs received as a result of this movement.

For example, let us consider the numerical example in Section 7.7.1, Scenario 1. The instant payoff for an attack on location 1 is found by letting the attacker attack location 1 and the defender defend location 2 when in state $s_1 = 000$. A successful attack would mean the movement to state $s_2 = 100$, hence showing that the defender has successfully attacked location 1. Assuming that this was an attack on the confidentiality of the resources at that location, the instant pay-off value would then be the value of $r_{C_{11}}$ in the placeholder matrix R_C. The remaining entries of this matrix can be found in similar ways.

The losses this attack causes the organisation and the way the system managers recover the location can then be calculated using loss functions and impulse-response functions as in Equations (6.5) and (4.14), respectively.

The fully integrated model can then be implemented into MATLAB and a sensitivity analysis can be performed to assess its validity. A series of thought experiments, similar to those carried out in Section 4, can then be carried out to analyse the findings to obtain a better understanding of the the validity and reliability of this model.

Once fully developed, real life attacks can be used in the model, such as the Conficker worm, to further assess its validity and reliability.

Much research has been carried out regarding the behaviour of the Conficker worm and how it spreads itself around a system. A detailed analysis of its distribution over networks, the probability of stealing information, its power of launching malicious services and the effectiveness of current reputation-based malware detection/warning systems is discussed by Shin et al. (2012). This information can be used, along with existing research regarding its behaviour, [164, 177], propagation pattern and victim distribution characteristics, [112], to obtain realistic probabilities of successful attack and vulnerability levels of locations within a system once the worm is inside the system, to set up a stochastic security game as described previously. The expected pay-offs obtained from this game can then be used as inputs of the placeholder matrix, as explained above, and its effects on an organisation analysed accordingly.

References

[1] S. Abramsky, *A domain equation for bisimulation*, Information and Computation, pp. 161-218, 1991.

[2] D. Acemoglu, M. Azarakhsh, and A. Ozdaglar. *Network security and contagion*. NBER Working Paper No. 19174, 2013.

[3] L. Aceto, W.J. Fokkink, C. Verhoef, *Structural operational semantics, in: Handbook of Process Algebra, North-Holland, Amsterdam*, 2001, pp. 197-292.

[4] A.Agah, S. K. Das, K. Basu, and M. Asadi, *Intrusion Detection in Sensor Networks: A Non-Cooperative Game Approach*, Proceedings of the 3rd IEEE International Symposium on Network Computing and Applications (NCA04), pp. 343-346, 2004.

[5] A.A.Akintola, G.A.Aderounmu and A.U.Osakwe, *Performing Modeling of an Enhanced Optimistic Locking Architecture for Concurrency Control in a Distributed Database System*, ACM vol.37, No.4, November 2005.

[6] Albany Software *Hardware Security Modules (HSMs)*, Albany Software Limited Company, 2012. [Online] Available at: `http://www.albany.co.uk/media/77808/hsms_final.pdf` [Accessed 14th Nov. 2017].

[7] F. Allen, and A. Babus: *Networks in Finance*. 2008.

[8] R. Allen, (2002) *Mathematical Modeling/Computationsl Science*, [Powerpoint Presentation], Available at: `http://slideplayer.com/slide/7351816/`, [Accessed: 2 Nov. 2017].

[9] T. Alpcan, T. Başar, *Network Security, A Decision and Game-Theoretic Approach*, Cambridge: Cambridge University Press. pp. 75-89, 2011.

[10] T. Alpcan, T. Başar, K.C. Nguyen, *Stochastic Games for Security in Networks with Interdependent Nodes*, In Proceedings of the IEEE International Conference on Game Theory for Networks (GameNets), 2009.

[11] *AlphaBay Also Offers Attraction Tickets and Luxury Goods*, AlphaBay Market. [online] Available at: `https://alphabaymarket.com/alphabay-offers-tickets-and-luxury-goods/` [Accessed 18 Nov. 2017].

[12] P. Ammann, D. Wijesekera, and S. Kaushik. *Scalable graph-based network vulnerability analysis*. In Proceedings of the 9th ACM Conference on Computer and Communications Security, pages 217-214. ACM, 2002.

[13] C. Anderson, and T. Moore. *The Economics of Information Security*. Science 314, 5799, p610, 2006.

[14] G. Anderson, and D. Pym. *A Calculus and Logic of Bunched Resources and Processes*. Theoretical Computer Science, 614:63-96, 2016.

[15] G.Anderson, M.Collinson, and D.Pym.*Utility-based Decision-making in Distributed Systems Modelling*. In Proc. TARK2013, Burkhard C. Schipper (editor), Chennai, 2013. Computing Research Repository (CoRR): `http://arxiv.org/corr/home`. ISBN: 978-0-615-74716-3.

[16] T. August and T. Tunca, *Network software security and user incentives*. Manage. Sci. 52, 11 (November), 1703-1720, 2006.

[17] M. Bacon, M. Rouse, (2014) *Social Engineering*, TechTarget, SearchSecurity, [online] Available at: `http://searchsecurity.techtarget.com/definition/social-engineering` [Accessed 23rd Oct. 2017].

[18] S. Bagchi, A. A. Clements, A. R. Hota, S. Sundaram *Optimal and Game-Theoretic Deployment of Security Investments in Interdependent Assets*. School of Electrical and Computer Engineering, Purdue Universtiy, GameSec 2016.

[19] J. d. Bakker and J. Zucker, *Processes and the denotational semantics of concurrency*, Information and Control, pp. 70-120, 1982.

[20] (Sept. 2017) *Which security investments make a difference?*, HELPNET SECURITY. [online] Available at: `https://www.helpnetsecurity.com/2017/09/27/security-investments/` [Accessed 18 Nov. 2017].

[21] A. Baldwin, I. Gheyas, C. Ioannidis, D.Pym, and J. Williams. *Contagion in Cybersecurity Attacks*. Workshop on the Economics of Information Security, 2012.

[22] N. Bambos, J. Mitchel, R.A. Miura-Ko, B. Yolken, *Security decision-making among interdependent organizations*, 21st IEEE Computer Security Foundation symposium, Stanford University, 2008b.

[23] N. Bambos, J. Mitchell, R.A. Miura-Ko, B. Yolken *Security investment games of interdependent organizations*. In: 2008 46th Annual Allerton Conference on Communication, Control, and Computing, pp. 252-260. IEEE (2008)

[24] N. Bambos, R. A. Miura-Ko, B. Yolken and Z. Zhou, *A game- theoretical formulation of influence networks*, in 2016 American Control Conference (ACC), pp. 3802-3807, IEEE, 2016.

[25] N. Bambos, P. Glynn, Z. Zhou, *Dynamics on Linear Influence Network Games Under Stochastic Environments*. Springer International Publishing, 114-126, 2016.

[26] T. Başar and Q. Zhu, *Dynamic policy-based IDS configuration*. In Proceedings of the 47th IEEE Con- ference on Decision and Control (CDC), 2009.

[27] K.J. Biba, *Integrity Considerations for Secure Computer Systems*, MTR-3153, The Mitre Corporation, June 1975.

[28] V.M. Bier and J. Zhuang. *Balancing terrorism and natural disasters-Defensive strategy with endogenous attacker effort*. Operations Research, 55(5): 976-991, 2007.

[29] M. Bishop, *Integrity Policies*, Powerpoint Slides, CS691 - Chapter 6.

[30] N. Blackmore (2015), *Fraudsters hacked emails to my solicitor and stole č340,000 from my property sale*, The Telegraph, May 2015. [Online] Available at: `http://www.telegraph.co.uk/finance/personalfinance/borrowing/mortgages/11605010/Fraudsters-hacked-emails-to-my-solicitor-and-stole-340000-from-my-property-sale.html`, [Accessed 10 Nov.2017]

[31] I. Blank, D. Crytser, (2013), *Classifying Linear Systems*, viewed 3rd February 2017, `https://www.math.ksu.edu/math240/math240.s06/m240classifysystems.pdf` .

[32] A. Bonaccorsi, P. Manfredi,A. Secchi. *Social heterogeneities in classical new product diffusion models*. LEM working paper 1991/21 SantAnna School for Advanced Studies, Pisa, 2004.

[33] M. Bowling and M. Veloso, *Rational and convergent learning in stochastic games*, in Proceedings of Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-01), pp. 1021-1026, 2001.

[34] J.R. Brown and Y. Fehige, *Thought Experiments*, The Stanford Encyclopedia of Philosophy (Summer 2017 Edition), Edward N. Zalta (ed.). [Online] Available at: `https://plato.stanford.edu/archives/sum2017/entries/thought-experiment` [Accessed 14 Nov.2017].

[35] S. Buchegger and T. Alpcan, *Security games for vehicular networks*. In Proceedings of the 46th Annual Allerton Conference on Communication, Control, and Computing. 244-251, 2008.

[36] CAIDA. Conficker/Conflicker/Downadup as Seen From the UCSD Network Telescope [Online]. Available: http://www.caida.org/research/security/ms08-067/conficker.xml

[37] C.C. Carmona. (2013) *A Review of Forecast Theory using Generalized Loss Functions*. Lecture notes, PDF.

[38] M. Cakanyildirim, W.T. Yue, Y.U. Rye, D. Liu, *Network Externalities, layered protection and IT security risk management*, Decision Support Systems, Elsevier, 2006.

[39] D.D. Caputo, S.L. Pfleeger, M.A. Sasse, P. Ammann, J. Offutt, L. Deng, *Barriers to Usable Security? Three Organizational Case Studies*. IEEE Security and Privacy, 14 (5) pp. 22-32, 2016.

[40] E. Cartwright, J. Hernandez-Castro, A. Stepanova, *To pay or not: Game theoretic models of Ransomware*, WEIS 2018.

[41] T. Caulfield and A. Fielder. *Optimizing time allocation for network defence*. In Journal of Cybersecurity, 2015.

[42] T. Caulfield, D. Pym, *Modelling and simulating systems security policy*, in: Proc. 8th SIMUTools, ACM Digital Library, 2015.

[43] T. Caulfield, C. Ioannidis, and D. Pym. *Discrete Choice, Social Interaction, and Policy in Encryption Technology Adoption* [Short Paper] In Proc. Financial Cryptography and Data Security, J. Grossklags and B. Preneel (editors), Lecture Notes in Computer Science, forthcoming, 2016.

[44] J. Cave, M. Klaver, V. Horvath, N. Robinson and A.P. Roosendaal *Data and Security Breaches and Cyber-Security Strategies in the EU and its International Counterparts*, European Parliament, Directorate-General for Internal Policies, Policy Department A: Economic and Scientific Policy A, Industry, Research and Energy, (2013). [online] Available at: `http://www.europarl.europa.eu/RegData/etudes/note/join/2013/507476/IPOL-ITRE_NT(2013)507476_EN.pdf` [Accessed on 9th October 2017]

[45] H. Cavusoglu, *Economics of IT Security Management.* In: Camp, L. and Lewis, S. (Eds), Economics of Information Security, Vol. 12, pp. 71-83. Springer US, 2004.

[46] H. Cavusoglu, H. Cavusoglu and S. Raghunathan, *Security patch management: Share the burden or share the damage.* Manage. Sci. 54, 4 (April), 657-670, 2008.

[47] *What is Certificate Transparancy*, Certificate Transparance. [online] Available at: `https://www.certificate-transparency.org/what-is-ct` [Accessed 14 Nov. 2017].

[48] C.Chang, S. Chen, and M. Wen. *Social Networks and Macroeconomic Stability*, Economics Discussion Papers, No 2013-4, Kiel Institute for the World Economy, 2013.

[49] R. Chicoisne, F. Ordonez. *Risk Averse Stackelberg Security Games with Quantal Response.* Springer International Publishing AG 2016, Q. Zhu et al. (Eds.) : GameSec 2016, LNCS 9996, pp. 83-100, 2016.

[50] A. Christmann and I. Steinwart. *Support Vector Machines*, Springer, XVI, Pages 21-47, 2008.

[51] M. Collinson and D. Pym. *Algebra and logic for resource-based systems modelling.* Mathematical Structures in Computer Science, 19:959-1027. doi:10.1017/S0960129509990077, 2009.

[52] M. Collinson, B. Monahan, and D. Pym. *Semantics for structured systems modelling and simulation.* In Proc. Simutools 2010. ACM Digital Library, ISBN 78-963-9799-87-5, 2010.

[53] M. Collinson, B. Monahan, D. Pym *A Discipline of Mathematical Systems Modelling*, Systems Thinking and Systems Engineering 2, Hewlett-Packard Development Company, L.P. and College Publications, 2012.

[54] M. Collinson, B. Monahan, and D. Pym. *A logical and computational theory of located resource. Journal of Logic and Computation* 19(6), 1207-1244, 2009. doi:10.1093/logcom/exp021.

[55] G. Coulouris, J. Dollimore, and T. Kindberg. *Distributed Systems; Concepts and Designs.* Addison Wesley, 3rd ed., 2000.

[56] M. Cremonini, D. Nizovtsev. *Understanding and influencing attackers' decisions: Implications for security investment strategies.* Workshop on the Economics of Information Security, Cambridge, UK, 2006.

[57] L. Criddle, *What is Social Engineering*, Webroot, [online] Available at: `https://www.webroot.com/gb/en/home/resources/tips/online-shopping-banking/secure-what-is-social-engineering` [Accessed 23rd Oct. 2017].

[58] Danezis, G. (2017). *The politics of the NHS WannaCrypt ransomware outbreak*. Bentham's Gaze. [online] Benthamsgaze.org. Available at: `https://www.benthamsgaze.org/2017/05/13/the-politics-of-the-nhs-wannacrypt-ransomware-outbreak/` [Accessed 2 Oct. 2017].

[59] C.J. DATE. *An introduction to database systems*. Addison Wesley. Reading, Fifth Edition,1991.

[60] R. Dewri, N. Poolsappasit, and I. Ray. *Dynamic security risk management using bayesian attack graphs*. Dependable and Secure Computing, IEEE Transactions on, 9(1):61-74, 2012.

[61] D. Duffie, M. Gustavo and S. Malamud. *Information Percolation with Equilibrium Search Dynamics*, Journal of the Econometric Society, Volume 77, Issue 5, pages 1513-1574, September 2009.

[62] N. DuPaul, *MAN IN THE MIDDLE (MITM) ATTACK*, CA Technologies, Veracode, APPSEC Knowledge Base. [Online]. Available at: `https://www.veracode.com/security/man-middle-attack` [Accessed 10 November 2017].

[63] C. Dwork, M. Naor, O. Reingold, G.N. Rothblum, and S. Vadhan. *On the complexity of differentially private data release: efficient algorithms and hardness results*. In Proceedings of the 41st annual ACM Symposium on the Theory of Computing, pages 381-390. ACM New York, NY, USA, 2009.

[64] G. Elliott and A. Timmermann. *Economic Forecasting*. Princeton: Princeton University Press, 2016.

[65] S. Roy, C. Ellis, S. Shiva, D. Dasgupta, V. Shandilya and O. Wu, *A survey of game theory as applied to network security,* In Proc. 43rd Hawaii International Conference on System Science, pp. 1-10, 2010.

[66] C. Eksin, A. Jadbabaie,P. Molavi, and A. Ribeiro. *Learning to coordinate in social networks* Oper. Res. 64(3), 605-621,2015.

[67] S. Esakkirajan, S. Sumathi, *Fundamentals of Relational Database Management Systems*, Warsaw: Springer, pp.564-565, 2007.

[68] F. Farahmand, S. Navathe, P. Enslow and G. Sharp *Managing vulnerabilities of information systems to security incidents.* In: ICEC '03 Proceedings of the 5th international conference on Electronic commerce, pp. 348-354. ACM: New York, USA, 2003.

[69] K.J. Farn, S.K. Lin, A.R.W. Fung, *A study on information security management system evaluation: assets, threat and vulnerability.* Computer Standards & Interfaces 2004; 26(6): 501e13, 2004.

[70] K. Faust. and W. Stanley. *Social Network Analysis: Methods and Applications.* Cambridge: Cambridge University Press, 1994.

[71] T.S. Ferguson, *Game Theory* Class notes for Math 167, Fall 2000.

[72] A. M. Fink. *Equilibrium in a stochastic n-person game.* Journal of Science in Hiroshima University, Series A-I, 28:89-93, 1964.

[73] W.J. Fokkink, *Introduction to Process Algebra*, Texts in Theoretical Computer Science, An EATCS Series, Springer, 2000.

[74] H. Fraser and M. Hauskrecht. *Planning Treatment of Ischemic Heart Disease with Partially Observable Markov Decision Processes.* Artificial Intelligence in Medicine, vol. 18, pp. 221-244, 2000.

[75] J. French D, Jerison and H. Miller. *Stability*, M.I.T., viewed 3rd February 2017, `http://math.mit.edu/~jorloff/suppnotes/suppnotes03/la8.pdf` .

[76] Gao et al, *Information Security Investment from the behaviour of a security provider*, 2015.

[77] X. Gao, M. Shue and W. Zhong. *Game-Theoretic Analysis of Information Sharing and Security Investment*, Journal of the Operational Research Society, Palgrave Macmillan, Volume 65, Number 11, Pages 1682-1691, Issue Number: 0160-5682, 2014.

[78] M. Gerber, R. Von Solms, *Management of risk in the information age.* Computers & Security 2005; 24(1):16e30, 2005.

[79] L. A. Gordon and M. P. Loeb . *The Economics of information Security*, University of Maryland, ACM Transactions on Information and Systems Security, Vol. 5, No. 4, November 2002, Pages 438-457, 2002.

[80] L.A. Gordon and M. P. Loeb. *Managing Cybersecurity Resources*. McGraw Hill, 2006.

[81] C.W.J. Granger, *Outline of Forecast Theory Using Generalized Cost Functions*, Spanish Economic Review, 1, 161-173, 1999.

[82] P. Grefen, R. de Vries, *A reference architecture for workflow management systems*, Data & Knowledge Engineering 27, Elsevier, 1997.

[83] J. Grossklags and B. Johnson, *Uncertainty in the weakest-link security game*. In Proceedings of the IEEE International Conference on Game Theory for Networks (GameNets). 673-682, 2009.

[84] K. A. Hansen, M. Koucký, N. Lauritzen, P. Bro Miltersen, and E. P. Tsigaridas, *Exact algorithms for solving stochastic games: extended abstract*. In STOC, pages 205-214, 2011.

[85] M. Hardt, G.N. Rothblum, and R.A. Servedio. *Private data release via learning thresholds*. Arxiv preprint arXiv:1107.2444, 2011.

[86] O. Hayatle and P. Taylor *A Markov Decision Process Model for High Interaction Honeypots*. Information Security Journal: A Global Perspective, 2013.

[87] G. Heal and H. Kunreuther, *Interdependent security*. In Journal of Risk and Uncertainty 26, 231 (2003).

[88] O.E. HedenStad, (2009) *Security Model for Resource Availability - Subject and Object type Enforcement*, Norwegian Defence Research Establishment, Powerpoint Slides. [Online] Available at: `https://wiki.uio.no/mn/ifi/AFSecurity/images/3/3c/AFSec200911-Hedenstad-FFI.pdf`. [Accessed at 20 Nov. 2017].

[89] C. Hennig and M. Kutlukaya, *Some thoughts about the design of loss functions*, REVSTAT-Statistical Journal, vol. 5, no. 1, 2007.

[90] M. Hennessy, *Algebraic theory of processes*, MIT Press, 1988.

[91] C.A.R. Hoare, *Communicating Sequential Processes*, Prentice-Hall International, London, 1985.

[92] S. Hwang, J. Knight, S.E. Satchell. *Forecasting Volatility Using Linex Loss Functions*. Financial Econometrics Research Center, Warwick Business School, 1999.

[93] INDEPENDENT SECURITY CONSULTANTS. (n.d.). *Threat, vulnerability, risk - commonly mixed up terms* - INDEPENDENT SECURITY CONSULTANTS. [online] Available at: `https://www.threatanalysis.com/2010/05/03/threat-vulnerability-risk-commonly-mixed-up-terms/` [Accessed 1 June 2017].

[94] C.Ioannidis, D. Pym, J. Williams. *Investments and Trade-offs in the Economics of Information Security*, Financial Cryptography and Data Security, 2009.

[95] C. Ioannidis, D. Pym, and J. Williams. *Fixed costs, investment rigidities, and risk aversion in information security: A utility-theoretic approach*. In Bruce Schneier, editor, Economics of Security and Privacy III. Springer, 171-192, 2012.

[96] C. Ioannidis, D. Pym, and J. Williams. *Information Security Trade-offs and Optimal Patching Policies*. European Journal of Operational Research, 216(2):434-444, 2012.

[97] ISO/IEC. *ISO/IEC 27002: code of practice for information security management*, 2005.

[98] M.O. Jackson. *Social and Economic Networks*. Princeton Universtiy Press, Princeton, 2008.

[99] A. Jaśkiewicz, A.S. Nowak, *Zero-sum stochastic games*. In: Başar, T., Zaccour, G. (Eds.), Handbook of Dynamic Game Theory. Springer International Publishing AG, 2016.

[100] A. Jaśkiewicz, A.S. Nowak, *Non-zero-sum stochastic games*. In: Başar, T., Zaccour, G. (Eds.), Handbook of Dynamic Game Theory. Springer International Publishing AG, 2017.

[101] S. Jha, O. Sheyner and J. Wing, *Two formal analysis of attack graphs*. In Computer Security Foundations Workshop, Cape Breton, Nova Scotia, Canada. IEEE, pp 49-63, 2002.

[102] Dr. Johnson, *Stability*, Introduction Multi-Step Methods Summary, School of Mathematics, Manchester University, Lecture notes. [Online] Available

at: http://www.maths.manchester.ac.uk/~pjohnson/CFD/Lectures/stability_handout.pdf [Accessed: 15th Nov. 2017]

[103] E. Jonsson and T. Olovsson, *A Quantitative Model of the Security Intrusion Process Based on Attacker Behavior*. IEEE Transactions on Software Engineering, 23(4), 1997.

[104] A. Kashyap, T. Başar and R. Srikant, *Correlated jamming on MIMO Gaussian fading channels*. IEEE Trans. Inform. Theory 50, 9, 2119-2123, 2004.

[105] https://www.theguardian.com/world/2017/feb/13/ex-gchq-whistleblower-attacks-plans-to-extend-dragnet-of-secrecy-act [Accessed 9th October 2017].

[106] https://www.theguardian.com/politics/2004/feb/26/interviews.iraq [Accessed 9th October 2017].

[107] C. Kelty (2011), *The Morris Worm*, Limn, Issue No. 1: Systemic Risk. [Online] Available at https://limn.it/the-morris-worm/, [Accessed 11th November 2011].

[108] G. Koop, M. H. Pesaran, S. M. Potter. *Impulse Response Analysis in Nonlinear Multivariate Models*. Journal of Econometrics, 74, 119-147, 1996.

[109] S. Kraus, J. Marecki, F. Ordonez,, P.Paruchuri, J. P. Pearce and M. Tambe. *Playing games with security: An efficient exact algorithm for Bayesian Stackelberg games*. In Proceedings of the Seventh International Conference on Autonomous Agents and Multiagent Systems, pages 895-902, 2008.

[110] A. Krause.(2010) *Advanced Topics in Machine Learning, Topic: Nonparametric learning and gaussian processes*. Lecture notes available from: http://courses.cms.caltech.edu/cs253/slides/cs253-14-GPs.pdf

[111] O. Kreidl. *Analysis of a Markov decision process model for intrusion tolerance*. In Dependable Systems and Networks Workshops, pages 156-161, 2010.

[112] J. Kristoff, *Experiences with Conficker C sinkhole operation and analysis*, in Proc. Australian Computer Emergency Response Team Conf.,Gold Coast, Australia, May 2009.

[113] K. Krol, J.M. Spring, S. Parking, M.A. Sasse, *Towards robust experimental design for user studies in security and privacy*. In: Proceedings of the 4th LASER

Workshop (2016): San Jose, US, 26 May 2016. IEEE: Piscataway, US. (In press), 2016.

[114] R. Lee and B. Wellman, *The New Social Operating System*, Cambridge, MA:MIT Press, 2012.

[115] T.H. Lee, (2007), "*Loss Functions in Time Series Forecasting*", Department of Economics, University of California, Riverside, Lecture Notes.

[116] Y.Liu,Q.Zhong,L.Chang,Z.Xia,D.He,C.Cheng, *A secure data backup scheme using multi-factor authentication*, IET Information Security,2016.

[117] M. Malheiros, C. Jennet, W. Seager, M.A. Sasse, *Trusting to learn: Trust and privacy issues in serious games.* In: Trust and Trustworthy Computing. (pp. 116 - 130). Springer: Berlin, 2011.

[118] M. Manshaei, Q. Zhu, T. Alpcan, T. Basar, and J.-P. Hubaux. *Game theory meets network security and privacy* ACM Computing Surveys, 2011.

[119] S. Mauw and M. Oostdijk. *Foundations of attack trees.* In Eighth Annual International Conference on Information Se- curity and Cryptology, LNCS. Springer, 2005.

[120] S. M. McCarthy, A. Sinha, M. Tambe and P. Manadhata, *Data Exfiltration Detection and Prevention: Virtually Distributed POMDPs for Practically Safer networks*, Springer International Publishing AG 2016, Q. Zhu et al. (Eds.): GameSec 2016, LNC9996, pp. 39-61, 2016.

[121] J. McDermott, *Attack-Potential-Based Survivability Modeling for High-Consequence Systems.* In Proc. of the Third IEEE Int. Information Assurance Workshop, Washington, DC, 2005.

[122] M. McDowell, *Understanding Denial-of-Service Attacks*, US-CERT, Security Tip (ST04-015). [Online] Available at: `https://www.us-cert.gov/ncas/tips/ST04-015` [Accessed 11th November 2017].

[123] I. Menache, and A. Ozdaglar. *Network games: theory, models and dynamics.* Synth. Lect. Commun. Netw. 4(1), 1-159, 2011.

[124] E. Miehling, M. Rasouli, and D. Teneketzis. *Optimal defense policies for partially observable spreading processes on bayesian attack graphs.* In Proceedings of the Second ACM Workshop on Moving Target Defense, pages 67-76. ACM, 2015.

[125] G. Milne and R. Milner, *Concurrent processes and their syntax*, J. Assoc. Comput. Mach., 26, pp. 302-321, 1979.

[126] R. Milner, *Communication and Concurrency*, Prentice Hall, New York, 1989.

[127] R. Milner, *Calculi for synchrony and asynchrony*. Theoretical Computer Science, 25(3):267-310, 1983.

[128] Mirror. (2017). *HATTON GARDEN JEWELLERY ROBBERY*. [online] Available at: `http://www.mirror.co.uk/all-about/hatton-garden-robbery` [Accessed 3 Oct. 2017].

[129] A. Morton, M.A. Sasse, *Privacy is a process, not a PET: a theory for effective privacy practice*. In: NSPW '12 Proceedings of the 2012 workshop on New Security Paradigms. (pp. 87 - 104). Association for Computer Machinery: New York, 2012.

[130] Murdoch, S., Sasse, A., Grossman, W. and Parkin, S. (2017). *Observing the WannaCry fallout: confusing advice and playing the blame game*Bentham's Gaze. [online] Benthamsgaze.org. Available at: `https://www.benthamsgaze.org/2017/05/19/observing-the-wannacry-fallout-confusing-advice-and-playing-the-blame-game/` [Accessed 2 Oct. 2017].

[131] J. F. Nash, Jr. *Equilibrium points in n-person games*, PNAS, 36:48-49, 1950.

[132] The Natural Sapphire Company, (2015). *The Graff Diamond Heist : One Of London's Biggest Robberies*. [online] The Natural Sapphire Company Blog. Available at: `https://www.thenaturalsapphirecompany.com/blog/the-graff-diamond-heist-the-most-expensive-in-londons-history` [Accessed 2 Oct. 2017].

[133] Neustar Security, *Worldwide DDoS Attacks & Cyber Insights Research Report*, DDoS Global Research Report, `https://ns-cdn.neustar.biz/creative_services/biz/neustar/www/resources/whitepapers/it-security/ddos/neustar-2017-worldwide-ddos-attacks-cyber-insights-research-report.pdf` [Accessed 9th October 2017].

[134] F. Nielson and Z. Aslanyan. *Pareto Efficient Solutions of Attack-Defence Trees*. in R Focardi and A Myers (eds), Principles of Security and Trust: Proceedings of the 4th International Conference, Springer, pp. 95-114, 2015.

[135] M.S. Olivier, *Database Privacy: Balancing Confidentiality, Integrity and Availability*. ACM SIGKDD Explorations of December 2002.

[136] R. Ortalo, Y. Deswarte and M. Kaâniche, *Experiments with Quantitative Evaluation Tools for Monitoring Operational Security*. IEEE Transactions on Software Engineering, 25(5):633-650, 1999.

[137] G. Owen, *Game Theory*, 3rd edn. New York, NY: Academic Press, 2001.

[138] A. Ozdaglar. *Network Games: Learning and Dynamics*, Conference on Decision and Control (CDC), Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, USA, 2008.

[139] D. Parker. *Fighting Computer Crime*. Wiley, MacMillan, ISBN: 978-0-471-16378-7, 1983.

[140] S. Pearson and A. Benameur, *Privacy, Security and Trust Issues Arising from Cloud Computing*, Proc. The 2nd International Conference on Cloud Computing 2010, IEEE, 2010, pp. 693-702, 2010.

[141] G. Plotkin, *A structural approach to operational semantics*, Report DAIMI FN-19, Computer Science Department, Aarhus University, 1981.

[142] B. Posey, M. Rouse, *Definition: Computer Exploit* , TechTarget, Search Security, `http://searchsecurity.techtarget.com/definition/exploit` [Accessed 18 October 2017].

[143] P. Prasad, B. Ojha, R.R. Shahi, R. Lal, *3-dimentional security in cloud computing*, in 3rd International Conference on Computer Research and Development (ICCRD, 2011), pp. 198-208, 2011.

[144] R. Publico (2017), *What is a Man-in-the-Middle Attack and How Can You Prevent It?*, GlobalSign GMO Internet Group, Global Sign Block. [Online] Available at: `https://www.globalsign.com/en/blog/what-is-a-man-in-the-middle-attack/`. [Accessed 10 November 2017].

[145] T. Quasim. "*Security Issues in Distributed Database System Model*, COMPUSOFT", An international journal of advanced computer technology, 2 (12), December-2013 (Volume-II, Issue-XII), 2013.

[146] R. Richardson, (2008) *CSI Computer Crime & Security Survey*, SURVEY, CSI Director. [Online] Available at: `http://www.sis.pitt.edu/jjoshi/courses/IS2150/Fall11/CSIsurvey2008.pdf`, [Accessed 20 Nov. 2017]

[147] S. Ross, *Introduction to Probability Models*. Eighth edition, Academic Press, 2003.

[148] B. Rossler. *The value of privacy*. Wiley, 2005.

[149] M. Rouse, M. Haughn, *Definition: Vulnerability*, TechTarget, WhatIs.com, http://whatis.techtarget.com/definition/vulnerability [Accessed 18 Oct. 2017].

[150] M. Rouse, *brute force cracking*, TechTarget, SearchSecurity, http://searchsecurity.techtarget.com/definition/brute-force-cracking [Accessed 5 Nov. 2017].

[151] M. Rouse, R. Richardson (2017), *ransomware*, TechTarget, SearchSecurity. [Online] Available at: http://searchsecurity.techtarget.com/definition/ransomware [Accessed 12th Nov. 2017].

[152] K. Safarzynska, J.C.J.M. van den Bergh, *Evolutionary models in economics: a survey of methods and building blocks*, Journal of Evolutionary Economics, 20 (3), pp. 329-373, 2010.

[153] K. Sallhammar, *Stochastic Models for Combined Security and Dependability Evaluation*, PhD Thesis, Norwegian University of Science and Technology, 2007.

[154] B. Schneier, *Secrets and Lies: Digital Security in a Networked World*. Wiley Publishing Inc, 2004.

[155] Seal, M. (2015). *Sony's Hacking Saga over The Interview; Seth Rogen and Evan Goldberg Speak Out*. [online] Vanity Fair. Available at: https://www.vanityfair.com/hollywood/2015/02/sony-hacking-seth-rogen-evan-goldberg [Accessed 17 Oct. 2017].

[156] Securedrop *Overview*. [online] Available at: https://docs.securedrop.org/en/stable/overview.html [Accessed 4 Oct. 2017].

[157] Selten, R. (1988). *A simple game model of kidnapping*. In Models of strategic rationality (pp. 77-93). Springer Netherlands.

[158] C Shapiro, H Varian, *Information Rules*, Harvard Business School Press, ISBN 0-87584-863-X, 1998.

[159] L. S. Shapley. *Stochastic games*. PNAS, 39:1095-1100, 1953.

[160] S. Shin, G. Gu, N. Reddy, and C. Lee. *A large-scale empirical study of Conficker.* Information Forensics and Security, IEEE Transactions on, 7(2):676-690, April 2012.

[161] R. Shokri, *Optimal user-centric data obfuscation*, ETH Zurich, 2014.

[162] D. Silver, *Lecture 2: Markov Decision Processes*, lecture notes, Reinforcement Learning, University College London, Delivered May 2015, 2015.

[163] E. H. Spafford, *The Internet Worm Incident*, Proceedings of the 1989 European Software Engineering Conference (ESEC 89); see also: Lecture Notes in Computer Science, vol. 87, Springer-Verlag, New York, 1989.

[164] SRI-International. *An analysis of Conficker C* [Online]. Available at `http://mtc.sri.com/Conficker/addendumC` [Accessed 23 Nov. 2017].

[165] M. Svorenová, M. Kwiatkowska, *Quantitative verification and strategy synthesis for stochastic games*, Eur. J. Control 30, 15-30, 15th European Control Conference, ECC16, 2016.

[166] L, Sweeney *k-anonymity: A Model for Protecting Privacy*. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems 10(5), 557-570, 2002.

[167] *A 'Kill Chain' Analysis of the 2013 Target Data Breach*, Majority Staff Report for Chairman Rockefeller,Committee on Commerce, Science and Transportation, United States Senate, (2014)`https://www.commerce.senate.gov/public/_cache/files/24d3c229-4f2f-405d-b8db-a3a67f183883/23E30AA955B5C00FE57CFD709621592C.2014-0325-target-kill-chain-analysis.pdf` [Accessed 9th October 2017]

[168] TechDifferences, *Differences between Client-Server and Peer-to-Peer Network*, 2017 `http://techdifferences.com/difference-between-client-server-and-peer-to-peer-network.html` [Accessed 17th Oct. 2017]

[169] Techopedia, *Threat* `https://www.techopedia.com/definition/25263/threat` [Accessed 18th Oct. 2017].

[170] Techopedia, *Attack* `https://www.techopedia.com/definition/6060/attack` [Accessed 18th Oct. 2017].

[171] C.W. Ten, G. Manimaran, and C.C. Liu, *Cybersecurity for critical infrastructures: Attack and defense modeling*, IEEE Trans. Syst. Man Cybern. A, Syst. Humans, vol. 40, no. 4, pp. 853-865, Jul. 2010.

[172] Techopedia, *Brute Force Attack* `https://www.techopedia.com/definition/18091/brute-force-attack` [Accessed 5th Nov. 2017].

[173] Theregister.co.uk. (n.d.). *74 countries hit by NSA-powered WannaCrypt ransomware backdoor: Emergency fixes emitted by Microsoft for WinXP+.* [online] Available at: `https://www.theregister.co.uk/2017/05/13/wannacrypt_ransomware_worm/` [Accessed 1 June. 2017].

[174] Tutorialspoint (2016). *Distributed DBMS Database Environments.* [online] Available at: `https://www.tutorialspoint.com/distributed_dbms/distributed_dbms_database_environments.htm` [Accessed 10 Feb. 2017].

[175] H. Varian. *A Bayesian Approach to Real Estate Assessment*, in S.E. Fienberg and A. Zellner (eds.), Studies in Bayesian Econometrics and Statistics in Honor of L.F. Savage, 195-208. Amsterdam: North-Holland, 1974.

[176] R. Von Solms and J. Van Niekerk, *From information security to cyber security*, Elsevier, Computers & Security, vol. 38, pp. 97-102, Oct. 2013.

[177] D. Watson, *Know Your Enemy: Containing Conficker* [Online]. Available at `https://www.honeynet.org/files/KYE-Conficker.pdf` [Accessed 23 Nov. 2017].

[178] Webopedia, *security vulnerability.* [online] `http://www.webopedia.com/TERM/S/security_vulnerability.html` [Accessed 1 June 2017].

[179] M.E. Whitman, H.J. Mattord, *Principles of information security.* 3rd ed. Thompson Course Technology; 2009.

[180] Wikipedia (2017), *Biba Model.* [Online] Available at: `https://en.wikipedia.org/wiki/Biba_Model` [Accessed 14 Nov. 2017].

[181] Wikipedia (2017). *Sony Pictures hack.* [online] Available at: `https://en.wikipedia.org/wiki/Sony_Pictures_hack` [Accessed 17 Oct. 2017].

[182] Wikipedia (2017) *AlphaBay.* [online] Available at: `https://en.wikipedia.org/wiki/AlphaBay` [Accessed 18 Nov. 2017].

[183] Wikipedia (2017), *Zero-Sum Game*. [online] Available at: `https://en.wikipedia.org/wiki/Zero-sum_game` [Accessed 23 Oct. 2017].

[184] Wikipedia (2017), *Attack (computing)* `https://en.wikipedia.org/wiki/Attack_(computing)` [Accessed 18 October 2017].

[185] Y. Wu, B. Wang, K. J. R. Liu, and T. C. Clancy,*Anti-jamming games in multi-channel cognitive radio networks*, IEEE Journal on Selected Areas in Communications, vol. 30, pp. 4-15, Jan. 2012.

[186] A. Zellner. *Bayesian Estimation and Prediction Using Asymmetric Loss Functions*, Journal of Forecasting, 8, 446-451,1986.

[187] K. Zetter (2017), *What is Ransomware? A Guide to the Global Cyberattack's Scary Method*, WIRED, Security. [Online] Available at: `https://www.wired.com/2017/05/hacker-lexicon-guide-ransomware-scary-hack-thats-rise/` [Accessed 12th Nov. 2017].

[188] B.M. Chain, I. J. Cox, J.C. Miller, S. Zhou, C. Zhang, *Optimizing Hybrid Spreading in Metapopulations*,, Scientific Reports, Statistical Physics, Thermodynamics and Nonlinear Dynamics, DOI: 10.1038/srep09924, 2015.

[189] Q. Zhu, C. Fung, R. Boutaba and T. Başar, *A game-theoretical approach to incentive design in collaborative intrusion detection networks.* In Proceedings of the International Conference on Game Theory for Networks (GameNets). 384-392, 2009.

[190] Q. Zhu and T. Başar, *Network security configuration: A nonzero-sum stochastic game approach.* In Proceedings of the American Control Conference (ACC), 2010.

[191] Q. Zhu, M. McQueen, C. Rieger and T. Başar, *Management of control system information security: Control system patch management.* In Proceedings of the Workshop on the Foundations of Dependable and Secure Cyber-Physical Systems (FDSCPS-11), 2011.

[192] S.A. Zonouz, H. Khurana, W.H. Sanders and T.M. Yardley *RRE: A game-theoretic intrusion response and recovery engine.* In Proceedings of the IEEE International Conference on Dependable Systems and Networks (DSN), 2009.

A Summary of Notations Used in Section 7

We provide a summary of the notations used in Section 7 with their description in the form of a table for the reader's convenience. For more clarity, the notations have been split into three different tables, one for notations used in the Markov Decision Process as described in Section 7.3 and one for those described in the Markov Security Game as described in Section 7.5 and one for additional notations introduced regarding influence models in Section 7.6.

Notation	**Description**
m	Total number of locations in a system
$N_S = 2^m$	Total number of states
S	Set of states (finite)
$s_1, s_2, ..., s_{N_S} \in S$	Different states
A	Set of actions (finite)
$P_a(s, s')$	Transition probability of moving from state s to s'
$R_a(s, s')$	Reward received when moving from state s to s'
γ	Discount factor
$\pi(s)$	Optimal policy to take action in state s
r_0^T	Partial reward
V_i^*	Value Function

Table A.1: Description of notations used in the Markov Decision Process as described in Section 7.3

Notation	**Description**
n	Total number of agents
A_i	Set of actions available to agent i
T	Transition function $S \times A \times S \to [0,1]$
R_i	Reward function for agent i
ρ	Mixed policy, $\rho : S \to PD(A_i)$, where PD =Probability Distribution
P^A	Attacker
P^D	Defender
N_A	Total number of attacker's actions available
N_D	Total number of defender's actions available
$A^A = \{a_1, ..., a_{N_A}\}$	Attacker action space
$A^D = \{d_1, ..., d_{N_D}\}$	Defender action space
$p^S = \{p_1, ..., p_{N_P}\}$	Probability distribution
$G(s(t)) = [G_{a,d}(s(t))]_{N_A \times N_D}$	Zero-sum game matrix
$q_{ij}(a,d)$	Probability of playing jth element when currently in element i
Q	Aggregate cost for defender
$p^A(s)$, $p^D(s)$	Strategy of attacker or defender, respectively

Table A.2: Description of additional notations used in the Markov Security Game as introduced in Section 7.5

Notation	**Description**
$\mathscr{N}$	Set of vertices in the network
$\mathscr{E}_s$	Set of edges in the network
$e_{ij} \in \mathscr{E}_s$	edges in the network
$\mathscr{G}_s\mathscr{N}, \mathscr{E}_s$	weighted directed graph of security assets containing a set of vertices and edges
$\mathscr{G}_v\mathscr{N}, \mathscr{E}_s$	weighted directed graph of vulnerabilties containing a set of vertices and edges
W_{ij}	Influence matrix
x	Independent Security Asset
y	Effective Security Asset
V	Vulnerability Matrix
$v_{ij} \in V$	Level of vulnerability of i due to j as a result of interdependencies in the system

Table A.3: Description of additional notations used in the Linear Influence models of the security assets and vulnerabilities in a sytem in Section 7.6