


國立臺灣大學資訊管理學研究所碩士論文

指導教授： 林永松 博士

達成資訊洩露程度最小化之  
近似最佳化防禦資源配置策略

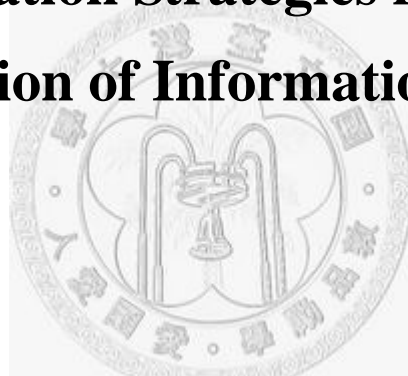


**Near Optimal Network Defense Resource  
Allocation Strategies for the  
Minimization of Information Leakage**

研究生： 曾中蓮 撰  
中華民國九十五年七月

達成資訊洩露程度最小化之  
近似最佳化防禦資源配置策略

**Near Optimal Network Defense Resource  
Allocation Strategies for the  
Minimization of Information Leakage**



本論文係提交國立台灣大學  
資訊管理學研究所作為完成碩士  
學位所需條件之一部份

研究生：曾中蓮 撰  
中華民國九十五年七月

## 謝誌

兩年的研究生活就要過去，而我的學生生涯也即將告一段落，看著投注許多心力的論文一步步成形，心中還真有股身為父母的喜悅及不捨；感謝這兩年來一直陪伴我的家人及朋友，沒有你們，這段路途我不會走的如此快樂。

首先要感謝的是我的父母曾令中以及梁丙樺，感謝你們讓我在愛與自由的環境中長大，使我可以選擇自己的人生道路，並在難過時得到溫暖的擁抱，讓我有勇氣面對一切難關；感謝我的弟弟曾中和，與我分享生活中的點點滴滴，並充當我的情緒垃圾桶及智囊團；也感謝我的哥哥與嫂嫂，在我研究受挫時為我打氣。

在這兩年的求學過程中，最要感謝的是林永松老師的指導。您總是不厭其煩的與我討論研究進度，並給我相當大的自由度，讓我能在研究中發揮創意；更重要的是，在跟隨老師的這兩年中，我學到許多待人處事應有的態度及道理，這份收穫是一輩子的禮物。另外，也要感謝交大資科林盈達教授、清大電機趙啟超教授、輔大資工呂俊賢教授以及本校孫雅麗教授在口試時對這篇論文的指正及建議，讓我從不同的角度思考這份研究，也使本論文更加嚴謹及完整。

特別感謝資安小組的各位成員：與我一起奮鬥兩年的義倫，我們終於闖關成功了!!俊維、承賓、雅芳及坤道，感謝你們不時給我加油及打氣，因為你們，資安小組變的充滿活力。已經畢業的建宏學長，感謝你以過來人的身分紓緩我心中的不安。最後是柏皓學長，你不只是領我進入資安界的師父，更是我最珍惜的朋友之一，感謝你在我喪氣時給我鼓勵，在我快樂時替我高興，這篇研究能夠完成都多虧了你。

壓軸的感謝要獻給文政、弘翕、勇誠及孝穎，你們讓這個實驗室變成了名符其實的 Happy Lab，在壓力緊繃的最後關頭，實驗室還能不時充滿歡笑，這都要歸功於你們。也感謝巧君、琳智學姊、書平學長、崑毅及翊恆在這段期間不時關心我的研究進度，讓我感到很窩心喔!!最後要感謝剛認識的國韋，多謝你提供有關賽局理論的資訊，讓我在口試前做了萬全準備。

最後，感謝我有這個機會進入台大，讓我除了在學術上有所發展外，也找到了人生想追求的事物，我想在台大的這兩年，將是我這輩子最難忘的回憶之一。

曾中蓮 謹識

于台大資訊管理研究所

民國九十五年七月

# 論文摘要

論文題目：達成資訊洩露程度最小化之近似最佳化防禦資源配置策略

作者：曾中蓮

九十五年七月

指導教授：林永松 博士

網際網路的普及和便利造成人們對網路的依賴，然而這也使得網路犯罪有機可乘。資訊竊取是造成最嚴重損失的網路犯罪之一，它不但造成金錢、財產之類的有形損失，還讓無形的企業及個人聲譽受損；因此如何幫助網際網路發展有效防禦策略，以降低資訊洩露的程度，就成了急需探討的研究議題。

在這篇論文中，我們將一個攻防情境轉化成二階的數學規劃問題；其中內層問題（AS 模型）敘述一個惡意攻擊者該如何配置其有限攻擊資源到目標網路，以竊取最多的機敏資訊，而在外層問題（DRAS 模型）中，目標網路的管理者則希望能有效配置其有限防禦資源，來將由資訊洩漏所引發的損失最小化。為了求得此問題的最佳解，我們採用以拉格蘭日鬆弛法為基礎的演算法來處理 AS 模型，而利用以次梯度法為基礎的演算法來處理 DRAS 模型。

關鍵詞：資訊竊取、拉格蘭日鬆弛法、數學規劃、網路攻防、網路存活度、最佳化、資源配置、無尺度網路

# THESIS ABSTRACT

GRADUATE INSTITUTE OF INFORMATION MANAGEMENT NATIONAL  
TAIWAN UNIVERSITY

NAME : LILLIAN, CHUNG-LIEN TSENG    MONTH/YEAR : JULY 2006

ADVISER : YEONG-SUNG LIN

## **Near Optimal Network Defense Resource Allocation Strategies for the Minimization of Information Leakage**

Dependency on the Internet is giving cyber criminals increasing opportunities to steal information. Information theft, one of the most damaging cyber-crimes, not only causes property damage and monetary loss to victims, it can also ruin their reputations. As a result, research into developing defense strategies against information theft on the Internet is a pressing need.

In this paper, we model an offence-defense scenario as a two-level mathematical programming problem. In the inner problem, defined by the AS model, an attacker allocates his limited attack power intelligently to the targeted network in order to steal as much valuable information as possible. Meanwhile, in the outer problem, defined by the DRAS model, the operator of the targeted network allocates limited defense resources appropriately to minimize the damage incurred by information theft. The Lagrangean relaxation-based algorithm is adopted to solve the AS problem, and a subgradient-based algorithm is proposed to solve the DRAS problem.

**Keywords: Information Theft, Lagrangean Relaxation, Mathematical Programming, Network Attack and Defense, Network Survivability, Optimization, Resource Allocation, Scale-free Networks**

# Table of Contents

謝誌.....	I
論文摘要 .....	II
THESIS ABSTRACT .....	III
Table of Contents .....	IV
List of Tables.....	VI
List of Figures.....	VII
<b>Chapter 1 Introduction.....</b>	<b>1</b>
<b>1.1 Background .....</b>	<b>1</b>
<b>1.2 Motivation.....</b>	<b>6</b>
<b>1.3 Literature Survey .....</b>	<b>8</b>
<b>1.3.1 Quantitative Analysis of Network Survivability .....</b>	<b>8</b>
<b>1.3.2 Scale-free Networks .....</b>	<b>11</b>
<b>1.4 Proposed Approach.....</b>	<b>14</b>
<b>1.5 Thesis Organization .....</b>	<b>15</b>
<b>Chapter 2 Problem Formulation of the DRAS and AS Models.....</b>	<b>17</b>
<b>2.1 Problem Description .....</b>	<b>17</b>
<b>2.2 Problem Formulation of the DRAS Model.....</b>	<b>19</b>
<b>2.3 Problem Formulation of the AS Model.....</b>	<b>26</b>
<b>Chapter 3 Solution Approach .....</b>	<b>29</b>
<b>3.1 Solution Approach for the AS Model .....</b>	<b>29</b>
<b>3.1.1 Lagrangean Relaxation Method .....</b>	<b>29</b>
<b>3.1.2 First-Stage Relaxation .....</b>	<b>33</b>
<b>3.1.2.1 Lagrangean Relaxation .....</b>	<b>33</b>
<b>3.1.2.2 The Dual Problem and the Subgradient Method.....</b>	<b>37</b>
<b>3.1.2.3 Getting Primal Feasible Solutions .....</b>	<b>38</b>
<b>3.1.3 Second-Stage Relaxation .....</b>	<b>40</b>
<b>3.1.3.1 Lagrangean Relaxation .....</b>	<b>41</b>
<b>3.1.3.2 The Dual Problem and the Subgradient Method.....</b>	<b>43</b>
<b>3.1.3.3 Getting Primal Feasible Solutions .....</b>	<b>44</b>
<b>3.1.4 Summary of the Solution Approach for the AS Model.....</b>	<b>47</b>
<b>3.1.4.1 Lagrangean Relaxation-based Algorithm.....</b>	<b>47</b>
<b>3.1.4.2 Initial Multiplier Determination.....</b>	<b>48</b>
<b>3.2 Solution Approach for the DRAS Model .....</b>	<b>49</b>
<b>Chapter 4 Computational Experiments .....</b>	<b>53</b>
<b>4.1 Computational Experiments with the AS Model .....</b>	<b>53</b>
<b>4.1.1 Simple Algorithm 1 .....</b>	<b>53</b>

4.1.2	Simple Algorithm 2 .....	54
4.1.3	Simple Algorithm 3 .....	55
4.1.4	Experiment Environment.....	56
4.1.5	Experiment Results.....	58
4.1.6	Discussion of Results.....	65
4.2	Computational Experiments with the DRAS Model .....	69
4.2.1	Experiment Environment.....	69
4.2.2	Experiment Results.....	70
4.2.3	Discussion of Results.....	75
Chapter 5	Conclusion and Future Work.....	77
5.1	Conclusion .....	77
5.2	Future Work .....	79
References	.....	83



## List of Tables

<b>Table 1-1 Survivability Definition Summary .....</b>	<b>4</b>
<b>Table 2-1 Problem Assumption and Description of the DRAS Model.....</b>	<b>21</b>
<b>Table 2-2 Given Parameters of the DRAS Model.....</b>	<b>23</b>
<b>Table 2-3 Decision Variables of the DRAS Model .....</b>	<b>23</b>
<b>Table 2-4 Given Parameters of the AS Model.....</b>	<b>26</b>
<b>Table 2-5 Decision Variables of the AS Model .....</b>	<b>27</b>
<b>Table 3-1 Heuristic_LR_1 Algorithm .....</b>	<b>40</b>
<b>Table 3-2 Relation between <math>y_i</math> and <math>\hat{a}_i(b_i)</math> .....</b>	<b>43</b>
<b>Table 3-3 Heuristic_LR_2 Algorithm .....</b>	<b>46</b>
<b>Table 3-4 LR Algorithm .....</b>	<b>47</b>
<b>Table 3-5 Heuristic_DRAS Algorithm .....</b>	<b>51</b>
<b>Table 3-6 Adjustment_Procedure Algorithm .....</b>	<b>52</b>
<b>Table 4-1 SA<sub>1</sub> Algorithm .....</b>	<b>53</b>
<b>Table 4-2 SA<sub>2</sub> Algorithm .....</b>	<b>54</b>
<b>Table 4-3 SA<sub>3</sub> Algorithm .....</b>	<b>56</b>
<b>Table 4-4 Experiment Parameter Settings for the AS Model .....</b>	<b>58</b>
<b>Table 4-5 Experiment Results of Medium-sized Networks (<math> N  = 100</math>).....</b>	<b>61</b>
<b>Table 4-6 Experiment Results of Large Networks (<math> N  = 400</math>).....</b>	<b>62</b>
<b>Table 4-7 Experiment Results of Extra-large Networks (<math> N  = 900</math>) .....</b>	<b>63</b>
<b>Table 4-8 Experiment Parameter Settings for the DRAS Model .....</b>	<b>70</b>
<b>Table 4-9 Experiment Results of Extra-small Networks (<math> N  = 25</math>).....</b>	<b>71</b>
<b>Table 4-10 Experiment Results of Small Networks (<math> N  = 49</math>) .....</b>	<b>72</b>
<b>Table 4-11 Experiment Results of Medium-sized Networks (<math> N  = 100</math>) .....</b>	<b>73</b>
<b>Table 5-1 Extra Notations Used in (IP 3).....</b>	<b>80</b>



## List of Figures

<b>Figure 1-1 Visualization of an Exponential Network [14] .....</b>	<b>12</b>
<b>Figure 1-2 Visualization of a Scale-free Network [14] .....</b>	<b>12</b>
<b>Figure 1-3 An Example of Power-law Distribution [19] .....</b>	<b>12</b>
<b>Figure 2-1 Initial State .....</b>	<b>20</b>
<b>Figure 2-2 Probing Neighbors .....</b>	<b>20</b>
<b>Figure 2-3 Attacking a Target .....</b>	<b>20</b>
<b>Figure 2-4 Post-attack Network State .....</b>	<b>20</b>
<b>Figure 2-5 Attack Tree .....</b>	<b>21</b>
<b>Figure 3-1 Concepts of the Lagrangean Relaxation Method .....</b>	<b>31</b>
<b>Figure 3-2 Lagrangean Relaxation Method Procedure .....</b>	<b>32</b>
<b>Figure 4-1 Susceptibility of Small Networks under Different Scenarios (<math> N  = 49</math>) .....</b>	<b>64</b>
<b>Figure 4-2 Susceptibility of Medium-sized Networks under Different Scenarios (<math> N  = 100</math>) .....</b>	<b>64</b>
<b>Figure 4-3 Susceptibility of Large Networks under Different Scenarios (<math> N  = 400</math>) .....</b>	<b>64</b>
<b>Figure 4-4 Susceptibility of Extra-large Networks under Different Scenarios (<math> N </math> <math>= 900</math>) .....</b>	<b>65</b>
<b>Figure 4-5 Susceptibility of Different Network Sizes and Topologies .....</b>	<b>65</b>
<b>Figure 4-6 Survivability of Extra-small Networks under Different Scenarios (<math> N </math> <math>= 25</math>) .....</b>	<b>74</b>
<b>Figure 4-7 Survivability of Small Networks under Different Scenarios (<math> N  = 49</math>) .....</b>	<b>74</b>
<b>Figure 4-8 Survivability of Medium-sized Networks under Different Scenarios (<math> N  = 100</math>) .....</b>	<b>74</b>

# Chapter 1 Introduction

## 1.1 Background

Because of the convenience and varied applications of the Internet, it has become an indispensable part of people's daily lives; however, increased dependence on the medium has also given cyber criminals more opportunities to steal information. Cyber-crime, which ranges from phishing and the use of botnets to information theft, has the potential to seriously disrupt our lives and even endanger our property. Therefore, the issue of how to deal with cyber-crime has become an urgent research topic in the field of network security.

Among various cyber-crimes, information leakage is one of the most serious threats because it jeopardizes network security and causes profound damage and loss. According to the CSI/FBI Computer Crime and Security Survey (2005) [1], theft of propriety information rates as one of the top three security incidents, resulting in dramatic loss to U.S. corporations and government agencies. This year, the survey [2] again indicated the great impact of information leakage. However, the loss and damage caused by information leakage are not as direct and explicit as those caused by DoS, DDoS, or viruses; instead, they are often only realized after the stolen information has been exploited. Moreover, some victims are unaware of an attack because the information stolen does not affect normal network operations. This "silent" attack behavior may not attract the victim's attention until the stolen

information is published or used, which could cause serious loss or damage to the victim and ruin his/her reputation. Consequently, network security experts have increased their efforts to develop strong countermeasures against information theft.

To prevent information theft by malicious attackers, network administrators and organizations have invested large amounts of resources, including money, time, and manpower, and deployed security hardware/software to strengthen their networks' robustness against attacks. However, due to the imperfection of software programming and communication protocols, malicious attackers always find ways to exploit the vulnerabilities of the Internet and launch attacks to compromise it. Given the inevitability of such attacks, perfect robustness of the Internet is unobtainable; hence, in recent years, the concept of security has been gradually generalized as an issue of *survivability* [3][4][5].

Typically, the states of network security are defined as *safe* or *compromised* [6]. However we believe that this binary concept is no longer sufficient to describe a system's states under malicious attack or random error conditions, because there is no attack-proof or error-free system in the world, especially as most systems are in unbounded environments [7]. The general concept of survivability describes how well a system can sustain normal service under abnormal conditions [8], and it complements the system states that are not covered by security. Thus, in this paper, we are particularly interested in the so-called *intermediate zone* between the safe and compromised states.

It is unfortunate that, despite the ongoing development of methods to strengthen network survivability, there is no consensus about a precise definition of the concept [9]. The most common definition [7] of survivability, proposed by Ellison et al. in 1999 [6], is: "the capability of a system to fulfill its mission, in a timely manner, in

the presence of attacks, failures, or accidents”. Table 1-1 lists several different definitions proposed by other researchers. Although the definitions are diverse, their underlying concepts can be generalized as four basic components [9]: **system**, the environment that provides services, the Internet for example; **usage**, the service requested by users; **minimum level of service**, a set of functional specifications for requested services, each of which has associated quality attributes and values; and **threats**, including random errors (accidental threats), malicious attacks (intentional threats), and catastrophic occurrences, such as natural disasters. Given the above components of survivability, we can analyze a system’s survivability quantitatively. The development and performance of previous quantitative analysis studies is discussed in Section 1.3.1.



**Table 1-1 Survivability Definition Summary**

No.	Researcher	Definition	Year	Origin
1	Deutsch and Willis	Survivability is the degree to which essential functions are still available, even though some part of the system is down.	1988	[10]
2	Liew and Lu	<p>Suppose the selected feature of the network is quantified and denoted by <math>x</math>. Survivability <math>S</math> is measured by the fraction of <math>x</math> that remains after the considered disaster has happened. The survivability can be characterized by following functions:</p> <ul style="list-style-type: none"> <li>• Expected survivability <math>E[S]</math> is the expected survivability value <math>s</math> after the disaster.</li> <li>• Worst-case survivability <math>s^w</math> is the minimum value of <math>s</math> under given disaster types.</li> <li>• <math>r</math>-percentile survivability <math>s_r</math> is the probability that <math>s</math> is no greater than <math>r\%</math> of the total resources.</li> <li>• Zero survivability <math>P_0</math> is the probability that <math>s</math> is 0.</li> </ul>	1992	[11]
3	Louca, Pitsillides, and Samaras	<p>(1) The ability of a network to maintain or restore an acceptable level of performance during network failure conditions by applying various restoration techniques.</p> <p>(2) The mitigation or prevention of service outage from potential network failures by applying preventative techniques.</p>	1999	[12]
4	Ellison, Fisher, and Linger	Survivability is the capability of a system (including networks and large-scale systems) to fulfill its mission, in a timely manner, in the presence of attacks, failures, or accidents.	1999	[6]

5	Knight and Sullivan	Survivability is the ability [of a system] to continue to provide one or more alternate services (possibly degraded, less dependable, or different) in a given operating environment when various events cause damage to the system or its operating environment.	2000 [3][4]
6	ANSI T1.523-2001, Telecom Glossary 2000 (revision of Federal Standard 1037C)	Survivability is a property of a system, subsystem, equipment, process, or procedure that provides a defined degree of assurance that the named entity will continue to function during and after a natural or man-made disturbance; e.g., a nuclear burst.  Note: For a given application, survivability must be qualified by specifying the range of conditions over which the entity will survive, the minimum acceptable level or post-disturbance functionality, and the maximum acceptable outage duration.	2000 [13]
7	T1A1.2 Network Survivability Performance Working Group	Suppose a measure of interest $M$ has the value $m_0$ just before a failure happens. The survivability behavior can be evaluated by the following attributes: <ul style="list-style-type: none"> <li>• <math>m_a</math> is the value of <math>M</math> just after the failure occurs.</li> <li>• <math>m_u</math> is the maximum difference between the value of <math>M</math> and <math>m_a</math> after the failure.</li> <li>• <math>m_r</math> is the restored value of <math>M</math> after some time <math>t_r</math>.</li> <li>• <math>t_R</math> is the time required for the system to restore the value <math>m_0</math>.</li> </ul>	2001 [8]

## 1.2 Motivation

Information leakage has been shown to be a serious threat to individuals, organizations, or even nations [1]; moreover, it is also recognized that it will become an increasingly critical security issue to organizations in the near future [2]. When important sensitive information is stolen, it is not just a matter of privacy invasion; in worst case scenarios, it may lead to property loss, financial ruin, and the loss of life if national security is involved. Even so, people tend to ignore threats that do not have a direct impact on their lives, and may gloss over the seriousness of such events when they do occur. Yet, once an incident is revealed or the stolen information is used or published, the damage and loss incurred may be inestimable. According to the 2005 Computer Crime and Security Survey by CSI/FBI [1], the average loss per U.S. organization caused by theft of proprietary information increased from \$168.5K in 2004 to \$355.5K in 2005; however, this is just the tip of the iceberg. Given the mushrooming losses resulting from information theft, an in-depth study of strategies against such attacks is indeed a pressing need.

Furthermore, unlike attackers in the past, who intended to crash a whole network or interrupt a system to stop it from providing normal services, attackers nowadays tend to exploit the vulnerabilities of a system and steal information from it, without necessarily crashing the system. Such information leakage often leads to huge damage and loss to the system's owner and the network operator. To prevent such occurrences, network operators must invest some resources to enhance the robustness of the whole network. However, resources are limited and, as already noted, it is impossible to make a network entirely attack-proof. Thus, the question arises: How can a network operator allocate his limited resources effectively, such that the extent of information leakage can be minimized?

To answer this question, we must begin by understanding the factors that make a network vulnerable. The resistance of each component against malicious attacks is one of the keys to a network's robustness. The stronger a component, the more effort an attacker must expend to compromise it. However, components in the network are not independent. On the AS (autonomous system) level of the Internet, a component's failure will lead to direct exposure of other components connected to it, and also increase the probability that they will be attacked. In this scenario, the topological structure of the network is an important factor that influences a network's robustness.

Recent studies have demonstrated that the Internet and many complex networks follow a power-law degree distribution, and are thus called scale-free networks [14]. Unfortunately, one of the main characteristics of scale-free networks is that they are highly susceptible to malicious attacks; that is, a network will almost certainly fail once a few of the most important components have been compromised [14]. Thus, the protection of such components is essential.

Nevertheless, knowing the factors that affect a network's resistance against attacks only gives us a hint about how to allocate defense resources, rather than a solution to the problem. In order to determine the best defense resource allocation strategy, we must first consider the best attack strategy. As the saying goes: "know your enemy, know yourself." Only by understanding how the attacker devises his strategy can the defender know how to protect the network. Therefore, in the following chapters we not only discuss how a network operator can allocate his defense budget optimally, but also how an attacker can adjust his strategies to steal as much information as possible in order to cause maximum damage to the network operator.

In addition, previous research shows that attempts to model attackers' actions in



an abstract, mathematical way and then predict the future actions of attackers based on those models is a non-trivial and unsolved issue [15]. Accordingly, to resolve this issue, we model the offense-defense game between an attacker and a network operator as a two-level mathematical optimization problem, and solve it with our proposed solution approaches. We also propose a new survivability measure that considers the level of damage incurred by an attacker.

## 1.3 Literature Survey

In this section, we review previous works on the quantitative analysis of network survivability and scale-free networks.

### 1.3.1 Quantitative Analysis of Network Survivability

Since the concept of survivability focuses on a system's behavior after failures, random errors or malicious attacks, good quantitative analysis measures for evaluating a network's post-failure survivability level are essential. Westmark [9] generalized quantitative measurements of network survivability into three categories: *connectivity*, *performance*, and *function of other quality or cost measures*. The first two measures are discussed below.

Network connectivity is defined as the minimum number of nodes or links that must be removed to disconnect an O-D (Origin-Destination) pair [13]. Generally, the more nodes or links needed to disconnect any O-D pair in a network, the more survivable the network will be. A network's connectivity can be calculated by finding the maximal amount of node-disjoint or edge-disjoint paths between each O-D pair in the network. For instance, Louca et al. [12] improve the survivability of a network by transforming it into a trellis graph and then find the  $K$ -best node-disjoint paths between a given O-D pair. Their proposed algorithm ensures that if  $k$  node-disjoint

paths exist between an O-D pair, they can be found exactly, and the total cost of  $k$  paths can be minimized. The algorithm can be applied to routing protocol designs to make a network fault-tolerant and minimize the impact of component failure.

To understand the impact of component failure on the Internet, many researchers have investigated the connectivity of the medium [14][16]. Different metrics are used to evaluate connectivity; for example, the maximal size of connected components, the average cluster size, the proportion of O-D pairs still connected, and the average network diameter. These researchers observed the Internet under different types of component failure scenarios, and found that node failure caused by malicious attacks is the major cause of Internet failure. This phenomenon is due to the topological structure of the Internet, which we discuss in the next section.

The second metric, network performance, is the level at which a network fulfills its QoS (Quality of Service) function [13]. Because the objective of a network is to provide satisfactory service, measures for analyzing network performance usually focus on evaluating the service quality that may be affected by failures, such as the number of functioning units, the number of connected nodes, the maximum traffic capacity, blocking probability, throughput, and the service restoration time [5]. According to the T1A1.2 working group on network survivability performance [8], the assessment of network survivability performance has two aspects: 1) the assessment of the frequency that abnormal conditions occur; and 2) the measurement of the impact of these conditions.

Most research into network performance focuses on the first aspect of performance assessment, and the most commonly used analysis technique is the continuous time Markov chain (CTMC) [5][17]. Liu and Trivedi [5] modeled network behavior as a truncated two-dimensional finite state system in which the state  $(i,j)$

indicates there are  $i$  available trunks and  $j$  of them carry ongoing flows. The transition between states is described by the arrival rate of service calls and the call holding rate. If  $i$  is equal to  $j$  (i.e., all trunks are carrying traffic), the new requested service will be blocked; hence, the performance of the network can be evaluated by the service blocking rate  $P_{bk}$ .

Keshtgary et al. [17] constructed a hierarchical survivability model, also known as a Markov chain model, which analyzes the availability of  $k$  disjoint paths between a given O-D pair. Each state of the model is a compound state of path sets. For example, state  $(P_1\bar{P}_2P_3)$  denotes that paths  $P_1$  and  $P_3$  are working and path  $P_2$  has failed. The transition rate between states is the probability of path failure or path restoration. In addition to the frequency of path failure, its impact is also analyzed. In this research, a Markov reward model (MRM) is proposed to evaluate the loss and the cost incurred by path failure and restoration. The total loss due to the unavailability of a path or paths and the capacity constraints on alternate paths when the primary path fails are defined as the *susceptibility* of the network; and the survivability of the network is calculated as  $(1 - \textit{susceptibility})$ .

In this paper, we focus on assessing the impact of malicious attacks. A novel survivability metric, which evaluates network performance by considering the total loss and damage resulting from information leakage, and the corresponding susceptibility metric are defined. Moreover, instead of using CTMC to analyze network survivability, we adopt mathematical programming with optimization techniques to accurately model the offense-defense scenario. Generally, the state transition in a Markov chain is two-way; however, only one-way transition exists in our scenario, since stolen information can not be redeemed. Thus, CTMC analysis is not really applicable in our research.

### 1.3.2 Scale-free Networks

The Internet has been appropriately described as a network with a complex topology, in which routers or domains are nodes, and connections between any two routers or domains are links [18][19]. Although previous research into complex networks showed that they can be described with the random graph model of Erdős and Rényi (ER model), Albert et.al [20] suggested that such networks can actually be divided into two major classes based on their connectivity distribution  $P(k)$ , given the probability that a node in the network is connected to  $k$  other nodes.

The first class of networks, *exponential networks*, is characterized by a  $P(k)$  that peaks at an average  $\langle k \rangle$  and decays exponentially for  $k \gg \langle k \rangle$ , also referred to as the Poisson distribution. The ER model and the small-world model of Watts and Strogatz (WS model) are the most well-known examples of exponential networks [14]. In contrast, *scale-free networks*, which include the Internet at the AS level [18][19] and the World Wide Web (WWW), proposed by Barabási and Albert [20], are characterized by a  $P(k)$  that decays as a power-law, i.e.,  $P(k) \sim k^{-\gamma}$ , free of a characteristic scale. If we plot the node degree and its cumulative distribution on a log-log axis, a straight line with slope  $-\gamma$  will be evident. Figures 1-1 and 1-2 are visualizations of two classes of network, and Figure 1-3 illustrates an example of the power-law distribution.

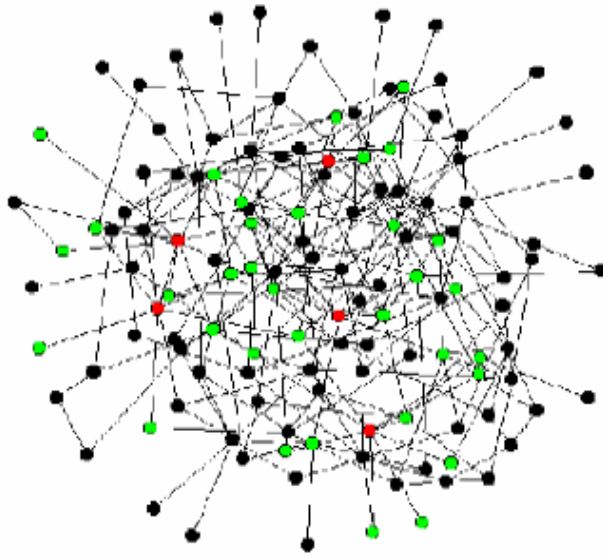


Figure 1-1 Visualization of an Exponential Network [14]

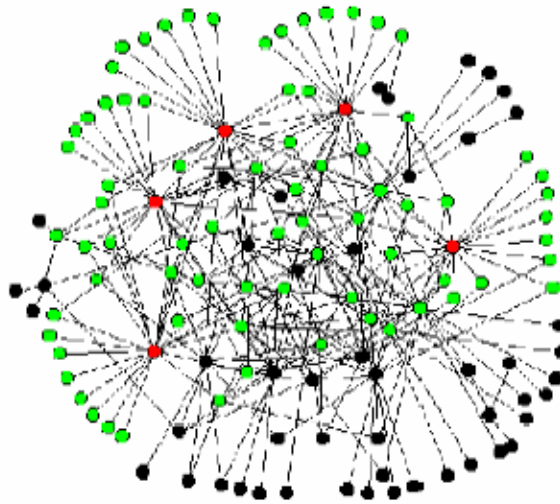


Figure 1-2 Visualization of a Scale-free Network [14]

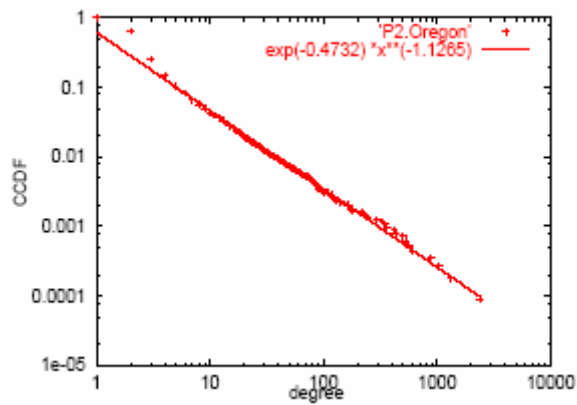


Figure 1-3 An Example of Power-law Distribution [19]

Albert et al. [14] observe that one of the major differences between exponential networks and scale-free networks is the homogeneity of nodes. Exponential networks result in homogeneous connectivity distribution, where each node has approximately the same number of links,  $k \approx \langle k \rangle$ . Inhomogeneity of scale-free networks, on the other hand, leads to the creation of highly connected nodes, which is practically impossible in exponential networks.

The power-law distribution of scale-free networks is generalized from two common characteristics of many real networks: *growth* and *preferential attachment* [14][20]. Most large networks in the real world evolve over time, e.g., social networks, citation networks, and the Internet. During a network's growth, newly added nodes are inclined to connect with existing nodes that have higher connectivity, which makes them susceptible to attack.

Although the inhomogeneous connectivity distribution of scale-free networks is more error-tolerant, it also has lower survivability against attacks [14]. The few existing highly connected nodes become a so-called "Achilles' heel", because once they have been compromised, information about the existence of most nodes in the network will be exposed. Unfortunately, in contrast to the uniform probability for each node in the case of random errors, nodes with higher connectivity, namely hubs, are much more likely to be targeted by an attacker. For this reason, the impact of random errors and that of malicious attacks on inhomogeneous scale-free networks are very different. Once the position of hubs becomes known, a scale-free network becomes highly susceptible to malicious attacks.

An even more serious problem, according to Park et al. [16], is that the Internet is more vulnerable than general scale-free networks due to its stronger preferential

attachment property, and the situation will get more serious over time. This phenomenon was also demonstrated by Faloutsos et al. In 1998, they discovered that the Internet's topology, both at the router level and the AS level, follows a power-law degree distribution [18]. In the following years, they continued to observe the evolution of the Internet [19], and found that the "six degrees of separation" property holds all the time. Similarly, Mahadevan et al.'s research [21] in 2006 also validated the power-law degree distribution and "six degrees of separation" phenomenon of the Internet AS-level topology. Since the size of the Internet grew from 3,000 nodes in 1998 to about 17,500 nodes in 2004, the findings implies that more and more nodes will suffer because of the compromise of few critical hubs. This inference not only supports the result of Park et al. but also highlights the urgent need for research into defense mechanisms to protect the Internet against malicious attacks.

## 1.4 Proposed Approach

In this paper, we propose a min-max mathematical model to describe the defense resource allocation strategy (DRAS) problem and the attack strategy (AS) problem precisely. By solving this two-level model optimally, we not only know the maximal potential damage that could be incurred under a certain defense budget allocation, but also find the best budget allocation strategy for the network administrator.

First, we formulate the DRAS problem as a mixed integer and linear programming (MILP) problem, where the problem objective is to minimize the potential total information value obtained by an attacker, subject to the network operator's budget limit. The potential total loss is derived from the result of the AS problem, which is formulated as another MILP problem. The objective of the AS problem is to maximize the total damage caused by information theft, subject to the

attacker's budget limit. We then propose using the Lagrangean Relaxation method, in conjunction with the subgradient method [22][23], to solve the AS problem. However, to solve the DRAS problem, a subgradient-based heuristic is proposed to adjust the defender's budget allocation strategy according to attacker's attack strategy.

A network's survivability is evaluated by the percentage of un-stolen information in the network. The higher the result, the more survivable the network is. In other words, a network with zero survivability would be fully compromised if the attacker allocates his attacker budget optimally. Comparisons of the survivability of networks under different defense budget allocation strategies and different topologies are presented in Chapter 4.

## **1.5 Thesis Organization**

The remainder of the thesis is organized as follows. In Chapter 2, MILP formulations of the DRAS and the AS problems are proposed. In Chapter 3, solution approaches to the AS problem and the DRAS problem are presented; in Section 3.1, solution approaches based on Lagrangean Relaxation are proposed; in Section 3.2, a solution approach to the DRAS problem based on the subgradient method is proposed. In Chapter 4, the computational results of the AS problem and the DRAS problem are presented. Finally, in Chapter 5, we present our conclusions and indicate possible directions of future research.





## **Chapter 2 Problem Formulation of the DRAS and AS Models**

### **2.1 Problem Description**

The problem we address is how a network operator should distribute a fixed amount of budget to each component so that the maximal damage and loss incurred by a potential attacker due to information leakage can be minimized. However, the “battle” between a network operator and an attacker is not static. A smart attacker will adjust his strategies dynamically to maximize the damage incurred, i.e., he will steal as much information as possible, if he knows the defense resource allocation strategy of the network operator and has sufficient attack power. It is therefore a challenge for network operators to derive adequate defense strategies against constantly changing attack strategies.

On the other hand, it is also difficult for an attacker to decide how to launch his attack. Just like the network operator, the attacker only has limited resources. Moreover, as it takes time and money to compromise a component, only part of the network can be compromised. Therefore, the resources must be fully utilized so that the attacker can gain the most valuable information and cause the maximum harm to the network operator.

Of course, the amount of information an attacker can gain from the network may differ when the defense resource allocation strategy changes. Hence, to evaluate the

efficiency of a certain defense budget allocation strategy, we analyze the survivability and the susceptibility of the network. The susceptibility metric, shown in the following equation, is defined as the percentage of stolen information, calculated by the percentage of the maximal damage incurred over the value of total information held by all the nodes in the network. The corresponding survivability metric is defined as the percentage of information not stolen, and which is a complement of the susceptibility metric. The more valuable the information stolen by the attacker, the lower the survivability of the network will be. Assume that  $d_i$  is the value of information contained by node  $i$ , where  $i \in N$ . Then, the metrics of network susceptibility and survivability can be presented as

$$Susceptibility(\%) = \left( \frac{\sum_{i \in \text{nodes that are compromised}} d_i}{\sum_{j \in \text{all nodes in the network}} d_j} \right) \times 100\%,$$

$$Survivability(\%) = (1 - Susceptibility) = \left( 1 - \frac{\sum_{i \in \text{nodes that are compromised}} d_i}{\sum_{j \in \text{all nodes in the network}} d_j} \right) \times 100\%,$$

respectively.

Using these metrics, we can evaluate the survivability of networks with different topological structures under the same defense budget allocation strategy, and the susceptibility of networks under different defense budget allocation strategies.

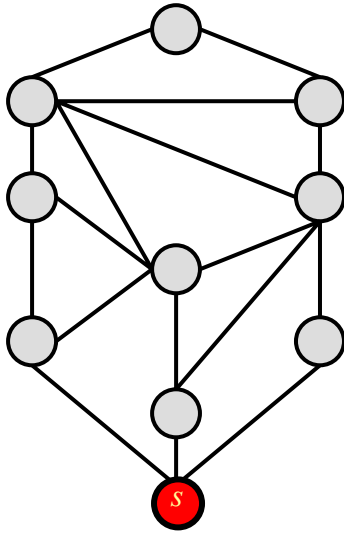
Note that the network we discuss here is the AS-level Internet. The Internet's topology is presented as an undirected graph, in which each node is a domain and each edge represents the inter-domain connection.

## 2.2 Problem Formulation of the DRAS Model

The evaluation of the robustness of a network under malicious attack is modeled as an optimization problem, where the objective of an attacker is to maximize the total damage incurred by compromising nodes in a network, while the defender tries to minimize the total damage. In the DRAS model, we assume that, like the defender, the attacker has complete information about the targeted network topology and the defense strategy. Although it is nearly impossible for an attacker to know everything about a network, we assume the worst case scenario for the network defender, so that the research is comprehensive. Because information theft is the goal of the attack, we only consider attacks on nodes, which are more common in the real world.

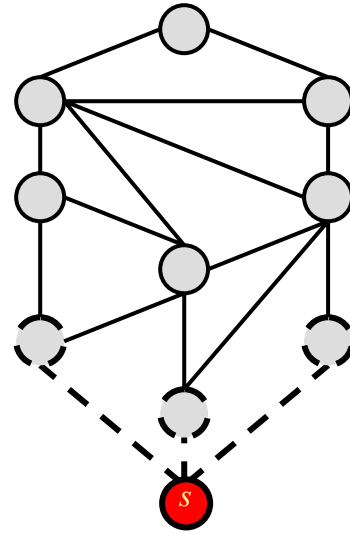
Initially, the attacker controls one node that connects directly to the targeted network, and that node is viewed as the initial hop-site to reach other nodes. Since the targeted network is at the AS level, the attacker cannot just attack any node directly. Instead, he can only reach uncompromised nodes from their immediate neighbors, which have already been compromised. Thus, the attacker needs to construct an *attack tree*, i.e., a tree consisting of compromised nodes and rooted at the initial hop-site.

To describe the attack procedures specifically, we adopt the following concept. First, the attacker occupies an initial node,  $s$  (Figure 2-1). He then adds all neighbors of  $s$  to the set of victim candidates (Figure 2-2). Next, he chooses a target from the candidate set and compromises it if he can apply enough attack power to it. The compromised node is used as a hop-site and its uncompromised neighbors are added to the set of victim candidates for the next stage of the attack (Figures 2-3 and 2-4). The attack ends when the total attack budget is consumed and an attack tree has been constructed (Figure 2-5). Diagrams of the attack behavior are presented below.



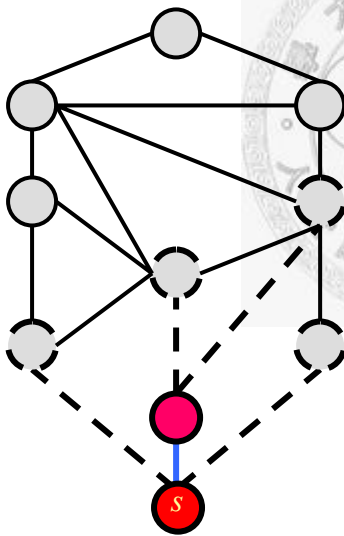
**Figure 2-1 Initial State**

Initially, the attacker is on node  $s$ .



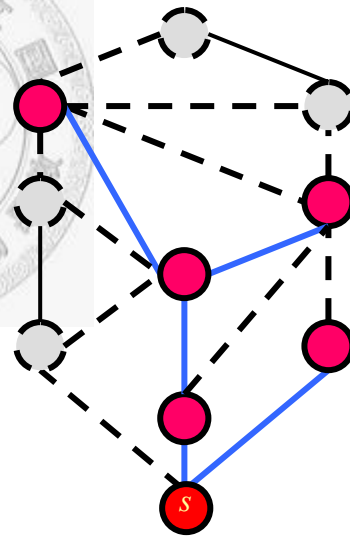
**Figure 2-2 Probing Neighbors**

Add uncompromised neighbors of the initial hop-site to the set of victim candidates.



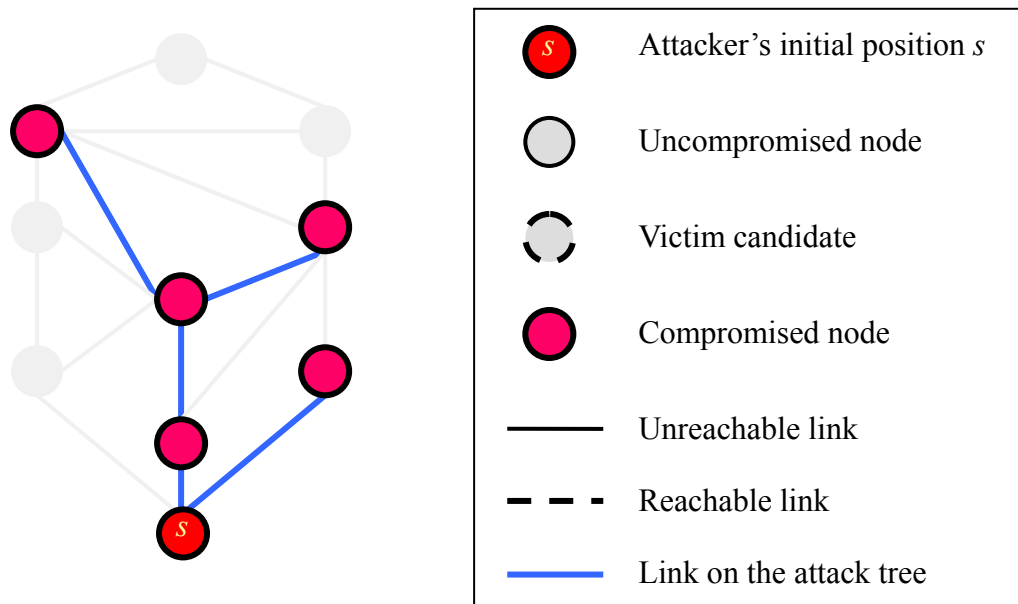
**Figure 2-3 Attacking a Target**

Compromising a node in the candidate set, and adding its uncompromised neighbors to the set.



**Figure 2-4 Post-attack Network State**

Continuing the attack until the attack resources are completely exhausted.



**Figure 2-5 Attack Tree**

The attack tree is constructed after the attack is completed.

The effort needed to compromise a node depends on the resources allocated to defend the node. Generally, the more defense resources a node has, the more robust it is, i.e., the attacker must use more resources to compromise that node. However, a node still has some defense capability even if no defense resources are allotted to it, since the node or the component itself is a shell for protecting the information. On the other hand, it should be noted that both the total defense and total attack resources are limited by their given budgets; therefore, how to distribute those resources effectively and intelligently is the objective of this work. The assumptions and description of the DRAS model are given in Table 2-1.

**Table 2-1 Problem Assumption and Description of the DRAS Model**

**Assumption**

- The attacker’s objective is to maximize the total damage by constructing an “attack tree” of the targeted network.
- The defender’s objective is to minimize the total damage by allocating a different budget to each node in the network.

- Both the attacker and the defender have complete information about the network topology.
- Both the attacker and the defender have resource budget limitations.
- Only node attacks are considered.
- Only malicious attacks are considered.
- Only AS-level networks are considered.
- A node is only subject to attack if a path exists from attacker's position to that node, and all the intermediate nodes on the path have been compromised.
- A node is compromised if the attack resources applied to the node are equal to or more than the defense capability of the node.

**Given**

- Defense resource budget  $B$
- Attack resource budget  $A$
- Damage  $d_i$  incurred by compromising node  $i$ , i.e., the value of information held by node  $i$
- Attacker's position  $s$ , which is connected to the target network
- The network topology and the network size

**Objective**

- To minimize the maximized total damage

**Subject to**

- The total defense cost must be no more than  $B$
- The total attack cost must be no more than  $A$
- The node to be attacked must be connected to the existing attack tree

**To determine**

- Defender: budget allocation strategy
- Attacker: which nodes to attack

We model the above problem as a min-max mathematical programming problem.

The parameters used in the model are defined in Table 2-2.

**Table 2-2 Given Parameters of the DRAS Model**

<b>Given Parameters</b>	
<b>Notion</b>	<b>Description</b>
$N$	The index set of all nodes in the network
$W$	The set of all O-D pairs, where the origin is node $s$ and the destinations are the nodes with positive $d_i$ , where $i, s \in N$
$d_i$	Damage incurred by compromising node $i$ , where $i \in N$
$P_w$	The index set of all candidate paths of an O-D pair $w$ , where $w \in W$
$A$	The total attack power
$B$	The total defense budget
$\delta_{pi}$	An indicator function, which is 1 if node $i$ is on path $p$ ; and 0 otherwise (where $i \in N, p \in P_w$ )

In this formulation, each node is given a positive  $d_i$  value, which is the value of the information it contains, and the damage incurred if it is compromised. The attacker's goal is to collect as much  $d_i$  as possible. The defender knows all the given parameters, but the attacker only has a priori knowledge of  $N$ ,  $A$ , and  $B$ .

The decision variables of the DRAS problem are listed in Table 2-3.

**Table 2-3 Decision Variables of the DRAS Model**

<b>Decision Variables</b>	
<b>Notion</b>	<b>Description</b>
$a_i$	Attack power applied to node $i$ , where $i \in N$
$b_i$	Budget allocated to protect node $i$ , where $i \in N$
$\hat{a}_i(b_i)$	The threshold of the attack power required to compromise node $i$ , i.e., the defense capability of node $i$ , where $i \in N$
$y_i$	1 if node $i$ is compromised; and 0 otherwise (where $i \in N$ )
$x_p$	1 if path $p$ is selected as the attack path; and 0 otherwise (where $p \in P_w$ )

The DRAS problem is then formulated as the following two-level MILP problem

(IP 1).



**Objective function:**

$$Z_{IP1} = \min_{b_i} \max_{y_i, a_i} \sum_{i \in N} d_i y_i \quad (\text{IP 1})$$

**Subject to:**

$$\sum_{w \in W} \sum_{p \in P_w} x_p \delta_{pi} \leq (|N| - 1) y_i \quad \forall i \in N \quad (\text{IP 1.1})$$

$$\sum_{p \in P_w} x_p = y_i \quad \forall i \in N, w = (s, i) \quad (\text{IP 1.2})$$

$$\sum_{p \in P_w} x_p \leq 1 \quad \forall w \in W \quad (\text{IP 1.3})$$

$$x_p = 0 \text{ or } 1 \quad \forall p \in P_w, w \in W \quad (\text{IP 1.4})$$

$$y_i = 0 \text{ or } 1 \quad \forall i \in N \quad (\text{IP 1.5})$$

$$0 \leq b_i \leq B \quad \forall i \in N \quad (\text{IP 1.6})$$

$$\sum_{i \in N} b_i \leq B \quad (\text{IP 1.7})$$

$$0 \leq a_i \leq \hat{a}_i(b_i) \quad \forall i \in N \quad (\text{IP 1.8})$$

$$\sum_{i \in N} a_i \leq A \quad (\text{IP 1.9})$$

$$\hat{a}_i(b_i) y_i \leq a_i \quad \forall i \in N. \quad (\text{IP 1.10})$$



**Explanation of the mathematical formulation:**

- Objective function: The objective is to minimize the maximized total damage  $\sum_{i \in N} d_i y_i$ . In the inner problem, an attacker tries to maximize the damage caused to the targeted network by deciding which nodes to attack, i.e., the  $y_i$  value of each node  $i$ . In the outer problem, the defender tries to minimize the damage caused by the attacker by allocating defense resources,  $b_i$ , to each node

appropriately.

- Constraint (IP 1.2) enforces that if a node is chosen for attack, i.e.,  $y_i = 1$ , the attacker must find a path between his initial position  $s$  and the targeted node.
- Constraint (IP 1.3) requires that if a node is chosen, the attack path for that node should be the only one.
- Constraint (IP 1.1) requires that a node can only be transited by  $(|N|-1)$  different attack paths, since there exist at most  $(|N|-1)$  targets. This constraint ensures the absence of a cycle on the attack tree, and also ensures that all nodes on each attack path are compromised.
- Constraints (IP 1.4) and (IP 1.5) limit the value of  $x_p$  and  $y_i$  to 0 or 1. Therefore, Constraints (IP 1.1) ~ (IP 1.5) jointly enforce that if a node is chosen for attack, there must be exactly one path from the attacker's initial position,  $s$ , to that node, and each node on the path must have been compromised. These constraints are jointly described as the “continuity constraints.”
- Constraints (IP 1.6) and (IP 1.7) restrict the amount of defense resources,  $b_i$ , that can be allocated to each node  $i$ . The total allotted defense resources,  $\sum_{i \in N} b_i$ , must not exceed the defense budget  $B$ .
- Constraints (IP 1.8) and (IP 1.9) restrict the attack power  $a_i$  that can be applied to each node  $i$ . The attack power cannot exceed the node's defense capability,  $\hat{a}_i(b_i)$  because it would be a waste of resources. Also, the total attack cost,  $\sum_{i \in N} a_i$ , must be less than the attack budget  $A$ .
- Finally, Constraint (IP 1.10) enforces that a node can only be compromised successfully if attack power applied to it is greater than its defense capability.

## 2.3 Problem Formulation of the AS Model

As noted earlier, it is extremely difficult to create a mathematical model that would predict an attacker's strategy. However, in the AS model we successfully formulate an attacker's behavior as an elegant mathematical optimization problem, which is also the inner problem of the DRAS model. By resolving this problem, we can predict the future actions of an intelligent attacker, and also design the best defense budget allocation strategy for a network operator. After the AS problem has been solved, its outcome is used as an input for the DRAS model to develop an advanced budget allocation strategy.

The model assumptions and attack processes of the AS model are the same as those of the DRAS model. We formulate the AS model as a mathematical maximization programming problem. The parameters are defined in Table 2-4.

Table 2-4 Given Parameters of the AS Model

Given Parameters	
Notion	Description
$N$	The index set of all nodes in the network
$W$	The set of all O-D pairs, where the origin is node $s$ ; and the destinations are the nodes with positive $d_i$ , where $i, s \in N$
$d_i$	Damage incurred by compromising node $i$ , where $i \in N$
$P_w$	The index set of all candidate paths of an O-D pair $w$ , where $w \in W$
$A$	The total attack power
$\hat{a}_i(b_i)$	The threshold of the attack power required to compromise node $i$ , i.e., the defense capability of node $i$ , where $i \in N$
$\delta_{pi}$	An indicator function, which is 1 if node $i$ is on path $p$ ; and 0 otherwise (where $i \in N, p \in P_w$ )

Note that  $\hat{a}_i(b_i)$ , which is a decision variable in the DRAS problem, is a given parameter in the AS problem. It is a function of  $b_i$ , the allotted budget of node  $i$ , and also the defense capability of the node. Node  $i$  can only be compromised if the

attacker applies more attack power than  $\hat{a}_i(b_i)$  to it.

The decision variables of the AS model and its formulation (IP 2) are given in Table 2-5.

**Table 2-5 Decision Variables of the AS Model**

Decision Variables	
Notion	Description
$a_i$	Attack power applied to node $i$ , where $i \in N$
$y_i$	1 if node $i$ is compromised; and 0 otherwise (where $i \in N$ )
$x_p$	1 if path $p$ is selected as the attack path; and 0 otherwise (where $p \in P_w$ )

**Objective function:**

$$Z_{IP2} = \max_{y_i, a_i} \sum_{i \in N} d_i y_i \equiv \min_{y_i, a_i} - \sum_{i \in N} d_i y_i \quad (\text{IP 2})$$

**Subject to:**

$$\sum_{w \in W} \sum_{p \in P_w} x_p \delta_{pi} \leq (|N| - 1) y_i \quad \forall i \in N \quad (\text{IP 2.1})$$

$$\sum_{p \in P_w} x_p = y_i \quad \forall i \in N, w = (s, i) \quad (\text{IP 2.2})$$

$$\sum_{p \in P_w} x_p \leq 1 \quad \forall w \in W \quad (\text{IP 2.3})$$

$$x_p = 0 \text{ or } 1 \quad \forall p \in P_w, w \in W \quad (\text{IP 2.4})$$

$$y_i = 0 \text{ or } 1 \quad \forall i \in N \quad (\text{IP 2.5})$$

$$0 \leq a_i \leq \hat{a}_i(b_i) \quad \forall i \in N \quad (\text{IP 2.6})$$

$$\sum_{i \in N} a_i \leq A \quad (\text{IP 2.7})$$

$$\hat{a}_i(b_i) y_i \leq a_i \quad \forall i \in N. \quad (\text{IP 2.8})$$

**Explanation of the mathematical formulation:**

- Objective function: The objective of the formulation is to maximize the total value of the information stolen. The result of the objective function is also the result of the inner problem in the DRAS model. For convenience, we transform (IP 2) from a maximization problem into an equivalent minimization problem. This does not affect the problem structure or the optimality conditions.
- Constraints (IP 2.1) ~ (IP 2.5) are the same as Constraints (IP 1.1) ~ (IP1.5) in the DRAS problem, and together form the “continuity constraints.”
- Constraints (IP 2.6), (IP 2.7), and (IP 2.8) are equal to Constraints (IP 1.8), (IP 1.9), and (IP 1.10) of the DRAS problem.
- The above formulation can be viewed as a 0-1 knapsack problem with continuity constraints, where each node represents an item, and the node’s information value and defense capability are the item’s profit and weight respectively. The total attack budget  $A$  is the total capacity of the knapsack, and the attacker tries to maximize the total profit up to the limit of the knapsack’s capacity.

## Chapter 3 Solution Approach

### 3.1 Solution Approach for the AS Model

#### 3.1.1 Lagrangean Relaxation Method

The Lagrangean relaxation method was first used to solve large-scale mathematical programming problems during the 1970s [24]. One of the method's basic concepts is “decomposition”; which efficiently reduces the complexities and difficulties of the primal problem. In fact, because of its efficiency and effectiveness in deriving proper solutions to many complicated programming problems, Lagrangean relaxation has become one of the most popular tools for solving optimization problems. Its applications include integer programming, linear programming combinatorial optimization, and non-linear programming problems. The method's performance is excellent, especially when dealing with large-scale mathematical programming applications [23].

The foundation of the Lagrangean relaxation method is to “pull apart” models by removing constraints and placing them in the objective function with associated Lagrangean multipliers ( $\mu$ ). The concept was inspired by the observation that many difficult integer programming problems arise from a relatively easy problem that is complicated by a set of side constraints. The Lagrangean relaxation method exploits this observation and creates a Lagrangean relaxation problem ( $LR_{\mu}$ ) in which the

complicating constraints are relaxed to the objective function by multiplying the corresponding  $\mu$  [23]. By transforming the primal problem (P) into a Lagrangean relaxation problem ( $LR_\mu$ ), we can decompose the complex mathematical model into several stand-alone subproblems, which can then be solved optimally by proper algorithms.

In addition, the Lagrangean relaxation method can provide us with some hints about obtaining the boundary of the objective function value. For a minimization optimization problem, the objective value,  $Z_D(\mu)$ , of the ( $LR_\mu$ ) is always a lower bound (LB) on the optimal solution of (P) [24]. In order to derive the tightest LB, we try to tune  $\mu$  to make  $Z_D(\mu)$  as large as possible, which is also known as the Lagrangean dual problem. The Lagrangean dual problem can be solved in various ways; the subgradient optimization technique is the most popular.

After resolving ( $LR_\mu$ ), we can examine the feasibility of the result for (P). If all the constraints in (P) are satisfied by the outcome, a primal feasible solution is found; otherwise, we need to develop proper heuristics to tune the infeasible solution to a feasible one. Furthermore, Lagrangean multipliers ( $\mu$ ) are also useful for adjusting the original heuristic to a Lagrangean-based modified heuristic, which may result in a better solution quality. Each feasible solution of (P) yields in an upper bound (UB) of the optimal value of (P); thus, the optimal solution to the primal problem is guaranteed to be within the Lagrangean LB and the primal feasible solution values. Figure 3-1 illustrates the main concepts of the Lagrangean relaxation method, and a detailed flow chart of Lagrangean relaxation method is presented in Figure 3-2.

In the following sections, we show how the AS problem is solved by the Lagrangean relaxation method, which consists of a two-stage relaxation procedure.

$$LB \leq \text{Optimal Objective Function Value} \leq UB$$

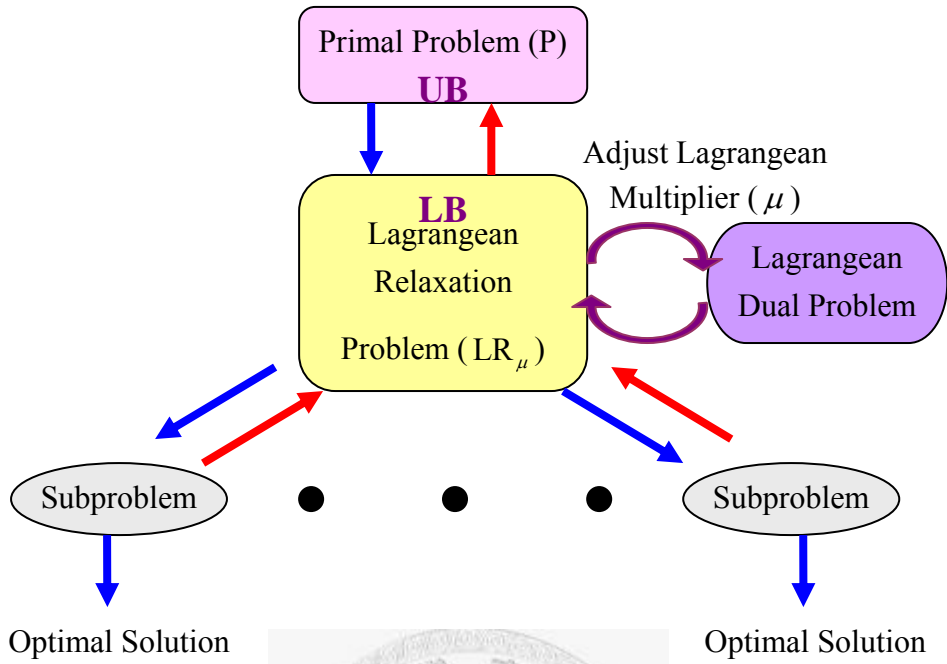
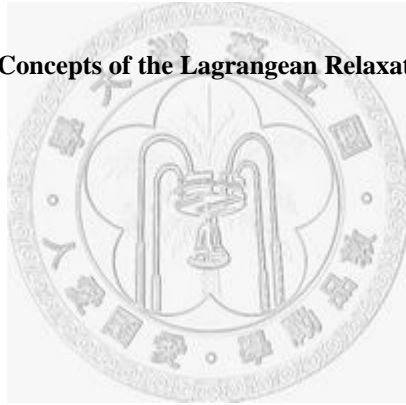


Figure 3-1 Concepts of the Lagrangean Relaxation Method





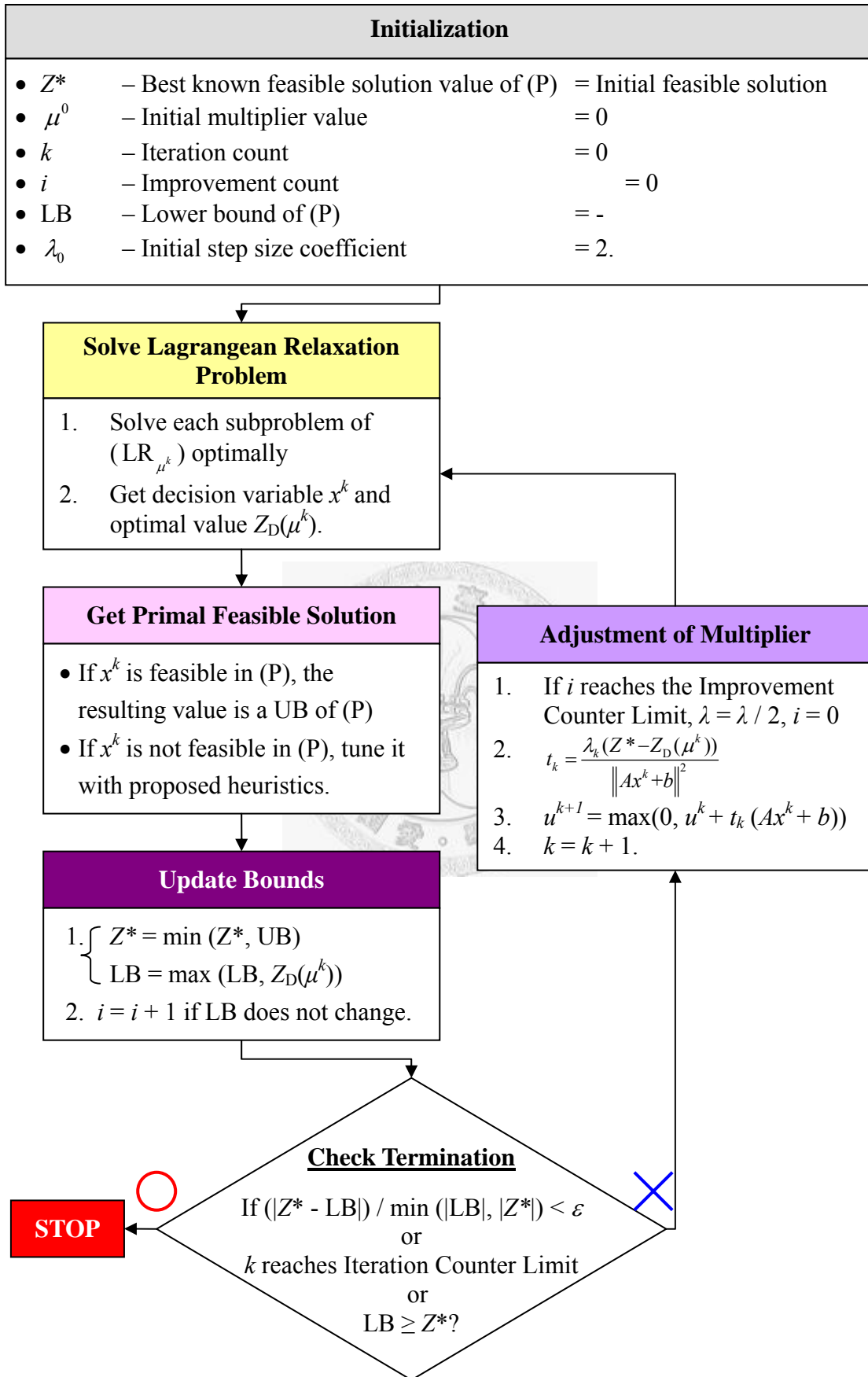


Figure 3-2 Lagrangean Relaxation Method Procedure

### 3.1.2 First-Stage Relaxation

In order to derive tighter UBs and LBs of the AS problem, we adopt a two-stage Lagrangean relaxation procedure. In the first stage, we relax three constraints of (IP 2), and construct a Lagrangean relaxation problem (LR 1). After solving (LR 1), the resulting UB and LB are taken as the initial UB and LB, respectively, of (IP 2) in the second stage, in which different constraints are relaxed.

#### 3.1.2.1 Lagrangean Relaxation

By applying the Lagrangean relaxation method, we transform the primal problem (IP 2) into the following Lagrangean relaxation problem (LR 1), where Constraints (IP 2.1), (IP 2.2), and (IP 2.8) are relaxed. With a vector of Lagrangean multipliers, the Lagrangean relaxation problem of (IP 2) is transformed as follows.

**Optimization problem:**

$$Z_D(\mu_1, \mu_2, \mu_3) = \min_{y_i} - \sum_{i \in N} d_i y_i + \sum_{i \in N} \mu_i^1 \left[ \sum_{w \in W} \sum_{p \in P_w} x_p \delta_{pi} - (|N| - 1) y_i \right] + \sum_{i \in N} \mu_i^2 \left[ \sum_{p \in P_{(s,i)}} x_p - y_i \right] + \sum_{i \in N} \mu_i^3 [\hat{a}_i(b_i) y_i - a_i] \quad (\text{LR 1})$$

**Subject to:**

$$\sum_{p \in P_w} x_p \leq 1 \quad \forall w \in W \quad (\text{LR 1.1})$$

$$x_p = 0 \text{ or } 1 \quad \forall p \in P_w, w \in W \quad (\text{LR 1.2})$$

$$y_i = 0 \text{ or } 1 \quad \forall i \in N \quad (\text{LR 1.3})$$

$$0 \leq a_i \leq \hat{a}_i(b_i) \quad \forall i \in N \quad (\text{LR 1.4})$$

$$\sum_{i \in N} a_i \leq A \quad (\text{LR 1.5})$$

The Lagrangean multipliers  $\mu_1, \mu_2$ , and  $\mu_3$  are the vectors of  $\{\mu_i^1\}, \{\mu_i^2\}, \{\mu_i^3\}$  respectively, in which  $\mu_1$  and  $\mu_3$  are non-negative and the variable  $\mu_2$  is unrestricted. To solve (LR 1), we decompose it into three independent and easily solvable optimization subproblems as shown below.

**Subproblem 1.1 (related to decision variable  $x_p$ )**

$$Z_{\text{Sub 1.1}}(\mu_1, \mu_2) = \min \sum_{i \in N} \sum_{w \in W} \sum_{p \in P_w} \mu_i^1 x_p \delta_{pi} + \sum_{i \in N} \sum_{p \in P_{(s,i)}} \mu_i^2 x_p \quad (\text{Sub 1.1})$$

**Subject to:**

$$\sum_{p \in P_w} x_p \leq 1 \quad \forall w \in W \quad (\text{Sub 1.1.1})$$

$$x_p = 0 \text{ or } 1 \quad \forall p \in P_w, w \in W. \quad (\text{Sub 1.1.2})$$

In this problem, we want to determine the value of  $x_p$  individually for each O-D pair. Note that Constraint (Sub 1.1.1) allows only one path to be chosen for an O-D pair. As described in the notations, each O-D pair  $w$  originates from an attacker's position  $s$  and ends at one target node  $i$ , where  $i \in N$ . Thus,  $\sum_{i \in N} \sum_{p \in P_{(s,i)}} \mu_i^2 x_p$  can be transformed into  $\sum_{w \in W} \sum_{p \in P_w} \mu_i^2 x_p + \sum_{p \in P_{(s,s)}} \mu_s^2 x_p$ , in which  $\sum_{p \in P_{(s,s)}} \mu_s^2 x_p$  can be ignored since no path starts and ends at the same node. After the transformation, we can further decompose (Sub 1.1) into  $|W|$  independent subproblems. For each O-D pair  $w = (s, i), i \in N$  and  $w \in W$ ,

$$Z_{\text{Sub 1.1}'}(\mu_1, \mu_2) = \min \sum_{p \in P_w} \left( \sum_{j \in N} \mu_j^1 \delta_{pj} + \mu_i^2 \right) x_p \quad (\text{Sub 1.1}')$$

**Subject to:**

$$\sum_{p \in P_w} x_p \leq 1 \quad \forall w \in W \quad (\text{Sub 1.1.1}')$$

$$x_p = 0 \text{ or } 1 \quad \forall p \in P_w, w \in W. \quad (\text{Sub 1.1.2'})$$

The algorithm for solving (Sub 1.1) is as follows:

**Step 1:** For each O-D pair  $w \in W$ , we find the minimum cost shortest path using  $\mu_j^1$  as the node weight by Dijkstra's minimum cost shortest path algorithm. The total cost of a path is the sum of the weights of the nodes on that path.

**Step 2:** For each O-D pair  $w \in W$ , we set the  $x_p$  value of each path  $p$  to zero except for the one already chosen to be the minimum cost shortest path for some O-D pair  $w$ , since not more than one path can exist between them.

**Step 3:** For each O-D pair  $w \in W$ , we examine the sum of its minimum path cost and the  $\mu_i^2$  value of its destination node. If the resulting value is non-positive, the  $x_p$  value of the minimum cost shortest path  $p$  between the O-D pair is set to one, because this is a minimization problem. The value of  $x_p$  is set to zero if its associated parameter is positive.

The time complexity of Dijkstra's algorithm is  $O(|N|^2)$ . Since the source of each path is the same, Dijkstra's algorithm only needs to be implemented once since its outcome is the minimum cost shortest path tree; thus, the total time complexity of (Sub 1.1) is  $O(|N|^2)$ .

**Subproblem 1.2 (related to decision variable  $y_i$ )**

$$Z_{\text{Sub 1.2}}(\mu_1, \mu_2, \mu_3) = \min \sum_{i \in N} (-d_i - \mu_i^1 (|N| - 1) - \mu_i^2 + \mu_i^3 \hat{a}_i(b_i)) y_i \quad (\text{Sub 1.2})$$

**Subject to:**

$$y_i = 0 \text{ or } 1 \quad \forall i \in N. \quad (\text{Sub 1.2.1})$$

(Sub 1.2) can be further decomposed into  $|N|$  independent sub problems, for which

we must decide the  $y_i$  value of each node  $i \in N$ . Since (Sub 1.2) is a minimization problem, and the value of each  $y_i$  is either zero or one, we can solve the problem by examining the associated parameters of  $y_i$  easily and optimally. For each node  $i \in N$ , if  $(-d_i - \mu_i^1(|N|-1) - \mu_i^2 + \mu_i^3 \hat{a}_i(b_i))$  is positive, the value of  $y_i$  is set to zero so that the value of this subproblem can be minimized. On the other hand, if the sum of the parameters is non-negative,  $y_i$  is set to one.

The time complexity of (Sub 1.2) is  $O(|N|)$ .

**Subproblem 1.3 (related to decision variable  $a_i$ )**

$Z_{\text{Sub 1.3}}(\mu_3) = \min \sum_{i \in N} (-\mu_i^3) a_i$	<b>(Sub 1.3)</b>
<b>Subject to:</b>	
$0 \leq a_i \leq \hat{a}_i(b_i)$	$\forall i \in N$ (Sub 1.3.1)
$\sum_{i \in N} a_i \leq A.$	(Sub 1.3.2)

By its nature, (Sub 1.3) is a fractional knapsack problem, in which the original maximized positive profit is replaced by minimized negative loss. To solve (Sub 1.3) optimally, we first sort each node  $i \in N$  by the parameter of each  $a_i$  and  $a_i$  itself in ascending order with  $(-\mu_i^3)$  as the primary key. Because of the non-negativity of  $\mu_i^3$ , the parameter of each  $a_i$  will be non-positive. Next, we check the array of sorted nodes from the left, and set the value of each  $a_i$  to  $\hat{a}_i(b_i)$ . We stop once the sum of  $a_i$  reaches  $A$ , or there is insufficient space to set the next  $a_i$  to  $\hat{a}_i(b_i)$ . In such a case, the next  $a_i$  is set to  $(A - \text{the summation of } a_i \text{ that have already been given a value})$ , and the remainder are set to zero.

The time complexity of (Sub 1.3) is  $O(|N|^2)$ .

### 3.1.2.2 The Dual Problem and the Subgradient Method

By solving the above subproblems optimally, the Lagrangean Relaxation problem (LR 1) can also be solved optimally. According to the weak duality theorem [24], for any set of the multipliers  $(\mu_1, \mu_2, \mu_3)$ ,  $Z_{D1}(\mu_1, \mu_2, \mu_3)$  yields an LB on  $Z_{IP2}$ . In the following, we construct a dual problem (D 1) to calculate the tightest LB and solve it by the subgradient method [22][23].

#### Dual Problem (D 1)

$$Z_D = \max Z_D(\mu_1, \mu_2, \mu_3) \quad (\text{D 1})$$

**Subject to:**  $\mu_1, \mu_3 \geq 0$ .

Let a vector  $m$  be a subgradient of  $Z_{D1}(\mu_1, \mu_2, \mu_3)$ . Then, in iteration  $k$  of the subgradient procedure, the multiplier vector  $\mu^k = (\mu_1^k, \mu_2^k, \mu_3^k)$  is updated by

$$\mu^{k+1} = \mu^k + t^k m^k,$$

where

$$m^k(\mu_1^k, \mu_2^k, \mu_3^k) = \left( \sum_{w \in W} \sum_{p \in P_w} x_p \delta_{pi} - (|N| - 1)y_i, \sum_{p \in P_{(s,i)}} x_p - y_i, \hat{a}_i(b_i)y_i - a_i \right);$$

and the step size,  $t^k$ , is determined by

$$t^k = \lambda \frac{Z_{IP2}^* - Z_D(\mu^k)}{\|m^k\|^2}.$$

$Z_{IP2}^*$  is the best UB on the primal objective function value found by iteration  $k$ . Note that  $\lambda$  is a scalar between 0 and 2. It is usually initiated with the value of 2 and halved if the best objective function value does not improve within a given iteration count.

### 3.1.2.3 Getting Primal Feasible Solutions

During first-stage Lagrangean relaxation, solutions to (LR 1) and their associated Lagrangean multipliers are considered in order to obtain a primal feasible solution for (IP 2). The concept of the proposed heuristic, denoted as Heuristic\_LR\_1, is described below.

Since an attacker's objective is to construct an attack tree, where the total value of the information gained is maximized, we develop the heuristic based on the greedy method. In the first step, we assign each node  $i$  a different weight,

$\max(0, \frac{\hat{a}_i(b_i) + |N|\mu_i^2}{(d_i)^2 + d_i/a_i})$ , where  $a_i$  is the solution obtained from (LR 1), and  $\frac{d_i}{a_i}$  is set

to zero if  $a_i$  is equal to zero. This formula reflects the ratio of the attack cost to the profit gained, i.e.,  $\frac{\hat{a}_i(b_i)}{d_i}$ ; the denominator is squared to stress the influence of the damage caused. Moreover, the formula also considers the hints obtained from the solutions to (LR 1). If non-zero attack power is applied to a node when solving (LR 1), the node is more likely to be chosen by the attacker when deriving primal feasible solutions.  $|N|\mu_i^2$  reflects the penalty of inconsistency between  $x_p$  and  $y_i$ , where a node is inclined to be targeted for attack if it has been chosen in (LR 1) but there is no attack path to it. After assigning the nodes' weights, we sort all nodes by their weights in ascending order for further processing by the greedy algorithm.

To apply the greedy method, we start by "activating" the first 50% of the nodes, starting from the node with the smallest weight. Note that a node can only be selected for attack if it has been activated. Then, using Prim's minimum cost spanning tree algorithm, a greedy-based algorithm, we try to construct a minimum cost sub-spanning tree with activated nodes from the attacker's initial position,  $s$ . Note that

the sub-spanning tree may not be complete, since the activated nodes may not form a connected graph.

Once the sub-spanning tree has been constructed, we examine each activated node in ascending order to see if it is on the sub-spanning tree, and if the total path cost from  $s$  to the node is affordable for the attacker. If the answer is positive, the node and all nodes on its path are compromised and added to the attack tree. Then the defense capability of each attacked node is deducted from the attacker's total resources. Next, the first half of the inactive nodes are activated, and Prim's algorithm is applied again to add new nodes to the previous sub-spanning tree. The procedure for activating nodes, constructing the sub-spanning tree, and then constructing the attack paths is repeated until the attacker does not have enough power to compromise any other node. At this time, the total profit gained by the attacker is a feasible solution to (IP 2).

The main idea of this heuristic arises from the attacker's intention that compromise nodes with smaller weights but moderate path costs for the most beneficial results. Thus, only attack paths that are composed of activated nodes, i.e. nodes with smaller weights, will be successfully constructed.

The total time complexity of Prim's algorithm is  $O(|N|\log|N|)$ . To activate all nodes in the network, the whole attack procedure needs to be repeated  $(\lceil \log |N| \rceil + 1)$  times. Thus, the total computational complexity of this heuristic is  $O(|N|\log^2|N|)$ .



**Table 3-1 Heuristic\_LR\_1 Algorithm**

```

//Initialization
FOR each node  $i$  {
     $weight = \max(0, \frac{\hat{a}_i(b_i) + |N|\mu_i^2}{(d_i)^2 + d_i/a_i})$ ;
}
Sort all nodes by their weights in ascending order;
Add source  $s$  to attack_tree;

//Construction of the attack_tree
WHILE ( $total\_attack\_cost < TOTAL\_ATTACK\_BUDGET$  AND there are still
uncompromised nodes) {
    Activate the first half of inactive nodes;
    Prim(); //construct the minimum cost sub-spanning tree rooted at  $s$ 
    FOR each activated and uncompromised node  $i$  {
         $path\_cost$  of  $i$  = summation of defense capability of all nodes on  $i$ 's path;
        IF ( $total\_attack\_cost + path\_cost$  of  $i \leq TOTAL\_ATTACK\_BUDGET$ ) {
            Compromise node  $i$  and all nodes on  $i$ 's path, and add them to the
            attack_tree;
             $total\_attack\_cost += path\_cost$  of node  $i$ ;
        }
    }
}

```

### 3.1.3 Second-Stage Relaxation

After the first-stage relaxation, we can get both a UB and a legitimate LB on the objective value of (IP 2). However, in order to narrow the range between the UB and LB, we need a second stage of relaxation to improve both the UB and LB. In the second stage, the initial UB and the initial LB are the best UB and the best LB of the first-stage relaxation respectively.

### 3.1.3.1 Lagrangean Relaxation

By applying Lagrangean relaxation method, we transform the primal problem (IP 2) into the following Lagrangean relaxation problem (LR 2), where Constraints (IP 2.1), (IP 2.2), and (IP 2.7) are relaxed. With a vector of Lagrangean multipliers, the Lagrangean relaxation problem of (IP 2) is transformed as follows.

**Optimization problem:**

$$Z_D(\nu_1, \nu_2, \nu_3) = \min_{y_i} - \sum_{i \in N} d_i y_i + \sum_{i \in N} \nu_i^1 \left[ \sum_{w \in W} \sum_{p \in P_w} x_p \delta_{pi} - (|N| - 1) y_i \right] + \sum_{i \in N} \nu_i^2 \left[ \sum_{p \in P(s,i)} x_p - y_i \right] + \nu^3 \left[ \sum_{i \in N} a_i - A \right] \quad (\text{LR2})$$

**Subject to:**

$$\sum_{p \in P_w} x_p \leq 1 \quad \forall w \in W \quad (\text{LR2.1})$$

$$x_p = 0 \text{ or } 1 \quad \forall p \in P_w, w \in W \quad (\text{LR2.2})$$

$$y_i = 0 \text{ or } 1 \quad \forall i \in N \quad (\text{LR2.3})$$

$$0 \leq a_i \leq \hat{a}_i(b_i) \quad \forall i \in N \quad (\text{LR2.4})$$

$$\hat{a}_i(b_i) y_i \leq a_i \quad \forall i \in N. \quad (\text{LR2.5})$$

The Lagrangean multipliers  $\nu_1$ , and  $\nu_2$  are the vectors of  $\{\nu_i^1\}$ ,  $\{\nu_i^2\}$  respectively, in which  $\nu_1$  is non-negative and the variable  $\nu_2$  is unrestricted. The Lagrangean multiplier  $\nu_3$  is non-negative. To solve (LR 2), we decompose it into three independent and easily solvable optimization subproblems, as shown below.

**Subproblem 2.1 (related to decision variable  $x_p$ )**

$$Z_{\text{Sub 2.1}}(v_1, v_2) = \min \sum_{i \in N} \left( \sum_{w \in W} \sum_{p \in P_w} v_i^1 x_p \delta_{pi} + \sum_{p \in P_{(s,i)}} v_i^2 x_p \right) \quad (\text{Sub 2.1})$$

**Subject to:**

$$\sum_{p \in P_w} x_p \leq 1 \quad \forall w \in W \quad (\text{Sub 2.1.1})$$

$$x_p = 0 \text{ or } 1 \quad \forall p \in P_w, w \in W. \quad (\text{Sub 2.1.2})$$

The subproblem is exactly the same as (Sub 1.1) in the first-stage relaxation; thus, we can adopt the algorithm proposed in Section 3.2.1.1 to solve (Sub 2.1) optimally.

The time complexity of (Sub 2.1) is  $O(|N|^2)$ .

**Subproblem 2.2 (related to decision variable  $y_i, a_i$ )**

$$Z_{\text{Sub 2.2}}(v_1, v_2, v_3) = \min \sum_{i \in N} (-d_i - v_i^1(|N| - 1) - v_i^2) y_i + v^3 \sum_{i \in N} a_i \quad (\text{Sub 2.2})$$

**Subject to:**

$$y_i = 0 \text{ or } 1 \quad \forall i \in N \quad (\text{Sub 2.2.1})$$

$$0 \leq a_i \leq \hat{a}_i(b_i) \quad \forall i \in N \quad (\text{Sub 2.2.2})$$

$$\hat{a}_i(b_i) y_i \leq a_i \quad \forall i \in N. \quad (\text{Sub 2.2.3})$$

This problem contains two decision variables,  $y_i$  and  $a_i$ , which are bound by Constraint (Sub 2.2.3). Their restricted relation is illustrated in Table 3-1. From Table 3-1, we can conclude that if only one variable is set to non-zero, the other can be set to a value other than zero. (Sub 2.2) can be further decomposed into  $|N|$  independent subproblems.

Table 3-2 Relation between  $y_i$  and  $\hat{a}_i(b_i)$

$y_i$ 's Value	$a_i$ 's Value
0	$[0, \hat{a}_i(b_i)]$
1	$\hat{a}_i(b_i)$

As  $y_i$  and  $a_i$  are independent of each other, we discuss them separately. First, the value of each  $y_i$  can be determined by the sum of its associated parameters, where  $i \in N$ . If the sum of the corresponding parameters of  $y_i$  is positive, it is set to zero; otherwise, it is *allowed* to be set to one.

Next, we consider  $a_i$ . Since this is a minimization problem, and  $\nu^3$  is a non-negative multiplier, it can only be minimized by setting all  $a_i$  to zero. However, due to the relation between  $a_i$  and  $y_i$ ,  $a_i$  must be set to  $\hat{a}_i(b_i)$  if  $y_i$ 's value is one. As a result, we need to consider the parameters of both  $a_i$  and  $y_i$  when determining the value of  $a_i$ .

For each node  $i \in N$  whose  $y_i$  is already set to zero, its  $\hat{a}_i(b_i)$  is also set to zero to comply with the limitations. For the other nodes, we examine the sum of the associated parameters of  $y_i$  and  $(\nu^3 \times \sum_{i \in N} a_i)$ . If the outcome is non-positive, the value of  $y_i$  is set to one determinately, and the value of  $a_i$  is set to  $\hat{a}_i(b_i)$ .

The time complexity of (Sub 2.2) is  $O(|N|)$ .

### 3.1.3.2 The Dual Problem and the Subgradient Method

By solving above subproblems optimally, the Lagrangean Relaxation problem (LR 2) can also be solved optimally. According to the weak duality theorem [24], for any set of the multipliers  $(\nu_1, \nu_2, \nu_3)$ ,  $Z_{D_2}(\nu_1, \nu_2, \nu_3)$  yields an LB on  $Z_{IP 2}$ . In the following, we construct a dual problem (D 2) to calculate the tightest LB and solve it by the subgradient method [22][23].

**Dual Problem (D 2),**

$$Z_D = \max Z_D(v_1, v_2, v_3) \quad (\text{D 2})$$

**Subject to:**  $v_1, v_3 \geq 0$ .

Let a vector  $m$  be a subgradient of  $Z_{D2}(v_1, v_2, v_3)$ . Then, in iteration  $k$  of the subgradient procedure, the multiplier vector  $v^k = (v_1^k, v_2^k, v_3^k)$  is updated by

$$v^{k+1} = v^k + t^k m^k,$$

where

$$m^k(v_1^k, v_2^k, v_3^k) = \left( \sum_{w \in W} \sum_{p \in P_w} x_p \delta_{pi} - (|N| - 1)y_i, \sum_{p \in P_{(s,i)}} x_p - y_i, \sum_{i \in N} a_i - A \right);$$

and the step size,  $t^k$ , is determined by

$$t^k = \lambda \frac{Z_{IP2}^* - Z_D(v^k)}{\|m^k\|^2}.$$

$Z_{IP2}^*$  is the best UB on the primal objective function value found by iteration  $k$ . Note that  $\lambda$  is a scalar between 0 and 2. Usually, it is initiated with the value of 2 and halved if the best objective function value does not improve within a given iteration count.

### 3.1.3.3 Getting Primal Feasible Solutions

To improve the solution quality of (IP 2), a heuristic is designed and implemented during the process of solving (LR 2), as in first-stage relaxation. In this heuristic, solutions to (LR 2) are adjusted to a feasible solution to (IP 2). The basic concept of the heuristic, denoted as Heuristic\_LR\_2, is described below.

In the problem assumption of the AS model, if a node is chosen to be compromised, the attacker must construct an attack path originating from the source  $s$

and ending at the targeted node. The union of all attack paths forms an attack tree. Based on this idea, we can utilize the solutions of (SUB 2.1), which is related to variable  $x_p$ . If the value of  $x_p$  is one, an attack path is constructed, and all nodes on the path are targeted. By taking the union of constructed attack paths, we can form an attack tree. If the total attack budget of the attack tree does not exceed the attacker's total budget, the total profit of the tree is a feasible solution to (IP 2), and the tree can be further expanded. Otherwise, we apply a recovery mechanism that recovers some of the compromised nodes. As shown in Section 4.1, the weight of each node is determined by  $\max(0, \frac{\hat{a}_i(b_i) + |N|\mu_i^2}{(d_i)^2 + d_i/a_i})$ .

In the first case, i.e., the total attack cost is less than the attacker's budget, the attack tree can be constructed a second time by using the remainder of the total attack budget. We start by building a spanning tree with Prim's algorithm, while retaining the original attack tree. Next, all the nodes that were not on the original attack tree are examined in ascending order according their weights. These nodes and all untaken nodes on their paths are then compromised if the attacker has sufficient budget. The total value gained by the attacker is a feasible solution to (IP 2).

In the second case, i.e., the total attack cost exceeds the attack budget, we recover the leaf node with the largest weight among all leaf nodes on the attack tree, and retrieve the attack budget. The recovery continues until the total attack cost is less than the total attack budget. Then, the total profit earned from the new attack tree is a feasible solution to (IP 2).

The time complexity of the first case is  $O(|N|\log|N|)$ , and that of the second case is  $O(|N|^2)$ ; therefore, the total computational complexity of this heuristic is  $O(|N|^2)$ .

Table 3-3 Heuristic\_LR\_2 Algorithm

```

//Initialization
FOR each node  $i$  {
     $weight = \max(0, \frac{\hat{a}_i(b_i) + |N|\mu_i^2}{(d_i)^2 + d_i/a_i})$ ;
}

//Take the union of attack_paths
FOR each attack_path  $p$  { //i.e., paths whose value of  $x_p$  is 1
    Each node  $i$  on  $p$  is added to the attack_tree;
     $total\_attack\_cost += defense\_power$  of node  $i$ ;
}

//Reconstruction of the attack_tree
IF ( $total\_attack\_cost < TOTAL\_ATTACK\_BUDGET$ ) {
    Prim(); //construct the minimum cost spanning tree on the basis of the attack_tree
    WHILE ( $total\_attack\_cost < TOTAL\_ATTACK\_BUDGET$  AND there are still
    uncompromised nodes) {
        Find node  $i$ , which is uncompromised AND whose weight is the smallest
        among all uncompromised nodes;
         $path\_cost$  of  $i$  = summation of defense capability of all nodes on  $i$ 's path;
        IF ( $total\_attack\_cost + path\_cost$  of  $i \leq TOTAL\_ATTACK\_BUDGET$ ) {
            Compromise node  $i$  and all nodes on  $i$ 's path, and add them to the
            attack_tree;
             $total\_attack\_cost += path\_cost$  of node  $i$ ;
        }
    }
}

//Recovery of compromised nodes
ELSE {
    WHILE ( $total\_attack\_cost > TOTAL\_ATTACK\_BUDGET$ ) {
        Find node  $i$ , which is a leaf_node of the attack_tree AND whose weight is the
        largest among all leaf_nodes;
        Recover node  $i$  and remove it from the attack_tree;
         $total\_attack\_cost -= defense\_power$  of node  $i$ ;
    }
}

```

## 3.1.4 Summary of the Solution Approach for the AS Model

### 3.1.4.1 Lagrangean Relaxation-based Algorithm

We propose a Lagrangean relaxation-based algorithm to solve the AS model and denote it as LR. This algorithm is based on the mathematical formulation of the AS model, i.e., (IP 2), as shown in Section 2.3. The relaxed problems are then solved optimally, as described in Sections 3.1.2 and 3.1.3, to get a LB for the primal problem. Next two heuristics are adopted to derive feasible solutions to the primal problem in Section 3.1.4, and a subgradient method is used to update the Lagrangean multipliers. As shown in Figure 3-2, the LR procedure is repeated iteratively until the stop condition is fulfilled. The time complexity of each iteration is  $O(|N|\log^2|N|)$ . Table 3-4 describes the complete LR algorithm for solving (IP 2).

Table 3-4 LR Algorithm

```
//Objective: maximize the total value of the information collected, i.e.,  $\min(-Z_{IP2})$ 
//Initialization of multipliers, as discussed in Section 3.1.5.2
Initialize the Lagrangean multiplier vectors  $(\mu_1, \mu_2, \mu_3)$  and  $(v_1, v_2)$  to be zero
vectors;
Initialize the Lagrangean multiplier  $v_3$  to be  $d_m / \hat{a}_m(b_m)$ ;
 $UB = 0$ ;  $LB = -TOTAL\_DAMAGE\_OF\_NETWORK$ ; //  $LB = -\sum_{i \in N} d_i$ 

improvement_counter = 0
 $\lambda = 2$ ; //step size coefficient
Init_Budget_Allocation_Strategy();

//Main LR procedure
FOR iteration = 1 TO ITERATION_COUNTER_LIMIT {
    IF iteration  $\leq$  (ITERATION_COUNTER_LIMIT / 2) {
        Solve (Sub 1.1);
        Solve (Sub 1.2);
        Solve (Sub 1.3);
         $Z^*_{IP2} = -Heuristic\_LR\_1()$ ; //due to the transformation of objective function
```



```

}
ELSE {
    Solve (Sub 2.1);
    Solve (Sub 2.2);
     $Z^*_{IP2} = -Heuristic\_LR\_2()$ ; //due to the transformation of objective function
}
Calculate  $Z_D$ ;

//Update bounds
IF ( $Z_D > LB$ ) {
     $LB = Z_D$ ;
    improvement_counter = 0;
}
ELSE {
    improvement_counter ++;
}
IF ( $Z^*_{IP2} < UB$ ) {
     $UB = Z^*_{IP2}$ ;
}

//Update step size and Lagrangean multipliers
IF improvement_counter = IMPROVEMENT_COUNTER_LIMIT {
    improvement_counter = 0;
     $\lambda = \lambda / 2$ ;
}
Update_Step_Size();
Update_Lagrangean_Multiplier();
}

```

### 3.1.4.2 Initial Multiplier Determination

In order to derive the tightest LB on  $Z_{IP2}$ , we must adjust the Lagrangean multipliers in dual problems (D 1) and (D 2) to maximize the objective function value of corresponding Lagrangean relaxation problems (LR 1) and (LR 2). Because the number of iterations in the LR procedure is limited, the initial value of the Lagrangean multipliers must be determined accurately, or the final LB will not

converge at a desirable point in time.

Usually the initial values of the Lagrangean multipliers are set to zero [22]; thus, the initial values of  $\mu_1$ ,  $\mu_2$ , and  $\mu_3$  in the first-stage relaxation are all zero. In the second-stage relaxation, different constraints are relaxed; however, multipliers  $\nu_1$  and  $\nu_2$  are the same as  $\mu_1$  and  $\mu_2$  due to the equality of their corresponding relaxed constraints. Therefore, the value of  $\mu_1$  and  $\mu_2$  at iteration  $(ITERATION\_COUNTER\_LIMIT) / 2$  can be considered as the initial values of  $\nu_1$  and  $\nu_2$ , as if  $\mu_1$  and  $\mu_2$  are still being used in the second-stage relaxation.

Specifically, the initial value of  $\nu_3$  is set to  $d_m / \hat{a}_m(b_m)$ , where  $m$  is the *critical item* and  $m \in N$ . When solving a fractional knapsack problem by the greedy method, only part of the *critical item*  $m$  is included in the knapsack. Since the AS model can be viewed as a 0-1 knapsack problem with continuity constraints, we can refer to Martello and Toth's research on 0-1 knapsack problems [26]. Following their research, the best LB for the objective function value of the AS model without continuity constraints can be obtained by  $\nu_3 = d_m / \hat{a}_m(b_m)$ . This approach is used in the design of the computational experiments, and the quality of LB is effectively improved.

### 3.2 Solution Approach for the DRAS Model

The outcome of the AS model indicates the result of the best attack strategy under a certain defense budget allocation strategy. As noted earlier, the main objective of the DRAS model is to minimize the total damage caused by an attacker when he/she tries to compromise a network. Thus, the optimal solution of the AS model can be used as the input of the DRAS model, in which we adjust the budget allocation strategy according to the current attack strategy. After the adjustment, we solve the AS model again and obtain another attack strategy corresponding to the new defense

budget allocation strategy. The interaction between attack strategies and defense strategies continues until a balance is reached.

The adjustment of the defense budget allocation strategy is based on the concept of the subgradient method, which adjusts each node's allotted budget according to the current step size. First, we examine the state of each node after the attack. If the node is undamaged, it implies that budget allocated to this node is too much or the reward of attacking this node is unprofitable. Either way, it suggests that the node has too much defense budget. Therefore, we deduct a small proportion of the budget from the uncompromised nodes, and allocate it to compromised nodes. The percentage deducted is equal to the step size coefficient, and is halved if the optimal solution of the DRAS model does not improve within a certain number of iterations.

Note that the exact amount of resources deducted from each node is different. Generally, the more times a node is used as a hop-site, the more important it is, since every time it is exploited, another node is compromised and extra damage is caused by the attacker. Thus, only small amount of budget is deducted from nodes that have been exploited frequently, even if they are not compromised under a certain defense resource allocation strategy. Furthermore, we propose an *impact factor* to normalize the number of times a node has been used as a hop-site. The factor is calculated by  $\frac{w_i}{w_{\max}}$ , where  $w_i$  is the average frequency that node  $i$  has been used as a hop-site, and  $w_{\max}$  is the potential maximal  $w_i$ , which is equal to the average number of nodes compromised during each attack. The higher the impact factor of a node, the lower the amount of resources that will be deducted from it, even if it is not attacked.

The complete heuristic for solving the DRAS model, denoted as Heuristic\_DRAS, and the core algorithm of the adjustment procedure, denoted

Adjustment Procedure, are presented below. The computational complexity of the Adjustment\_Procedure is  $O(|N|)$ .

**Table 3-5 Heuristic\_DRAS Algorithm**

```

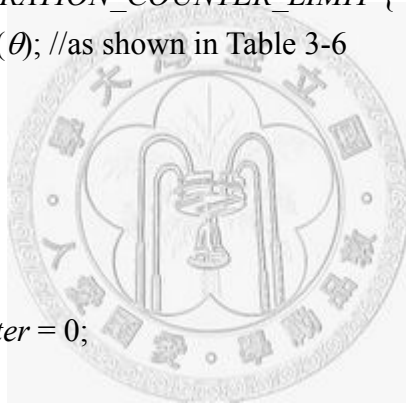
//Objective: minimize the maximized total damage, i.e., min max  $Z_{IP\ 1}$ 
//Initialization
Init_Budget_Allocation_Strategy();
UB = -LR(); //the return value of LR() is negative due to the objective function
transformation in the AS model
improvement_counter = 0
improvement_stage_counter = 0;
 $\theta = 0.5$ ; //initial step size coefficient

//Main Heuristic_DRAS procedure
FOR iteration = 1 TO ITERATION_COUNTER_LIMIT {
    Adjustment_Procedure( $\theta$ ); //as shown in Table 3-6
     $Z^*_{IP\ 1} = -LR()$ ;

    //Update UB
    IF ( $Z^*_{IP\ 1} < UB$ ) {
        UB =  $Z^*_{IP\ 1}$ ;
        improvement_counter = 0;
    }
    ELSE {
        improvement_counter ++;
    }

    //Update step size
    IF improvement_counter = IMPROVEMENT_COUNTER_LIMIT {
        improvement_counter = 0;
        improvement_stage_counter ++;
         $\theta = \theta / 2$ ;
    }
}

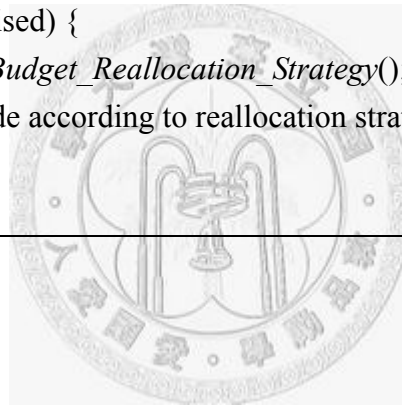
```



**Table 3-6 Adjustment\_Procedure Algorithm**

```
//Initialization
total_defense_cost = 0;
FOR each node  $i$  {
    IF (node  $i$  is uncompromised) {
         $b_i = b_i(1 - \theta(1 - \frac{w_i}{w_{\max}}))$ ; //  $b_i$  is the defense budget of node  $i$ ,  $w_i$  is the average
        number of times node  $i$  is used as a hop-site
    }
    total_defense_cost +=  $b_i$ ;
}
collection = TOTAL_DEFENSE_BUDGET - total_defense_cost

//Reallocation of defense budget
FOR each node  $i$  {
    IF (node  $i$  is compromised) {
         $b_i += collection * Budget\_Reallocation\_Strategy()$ ; //reallocate spare budget to
        compromised node according to reallocation strategy
    }
}
```



## Chapter 4 Computational Experiments

### 4.1 Computational Experiments with the AS Model

To demonstrate that our proposed heuristics are effective we implement the following two simple algorithms for comparison purposes.

#### 4.1.1 Simple Algorithm 1

The concept of simple algorithm 1 is derived from the heuristic of first-stage Lagrangean relaxation shown in Section 3.1.4.1. Hence, simple algorithm 1 also adopts the concept of the greedy method whereby the node with smallest weight is activated first and is a priority attack target if its path cost is acceptable. However, unlike the proposed heuristic, we use  $\frac{\hat{a}_i(b_i)}{(d_i)^2}$ , as the weight of each node  $i$  in the network. The pseudo code of simple algorithm 1, denoted as SA<sub>1</sub>, is presented below.

Table 4-1 SA<sub>1</sub> Algorithm

```
//Initialization
FOR each node  $i$  {
     $weight = \frac{\hat{a}_i(b_i)}{(d_i)^2}$ ;
}
Sort all nodes by their weight in ascending order;
Add source  $s$  to the attack_tree;

//Construction of attack_tree
WHILE ( $total\_attack\_cost < TOTAL\_ATTACK\_BUDGET$  AND there are still
uncompromised nodes) {
```

```

Activate the first half of inactivated nodes;
Prim(); //construct the minimum cost spanning tree rooted at s
FOR each activated and uncompromised node  $i$  {
    IF ( $total\_attack\_cost + path\_cost$  of  $i \leq TOTAL\_ATTACK\_BUDGET$ ) {
        Compromise node  $i$  and all nodes on  $i$ 's path, and add them to the
         $attack\_tree$ ;
         $total\_attack\_cost += path\_cost$  of node  $i$ ;
    }
}

```

### 4.1.2 Simple Algorithm 2

The concept of simple algorithm 2 is derived from the idea that nodes with smaller weights have a higher priority to be attacked. Here, we adopt Prim's algorithm to predetermine the path from  $s$  to each node. Then, the uncompromised node with the smallest weight is targeted, and its attack path is constructed if the attacker has sufficient attack power. As in simple algorithm 1, we use  $\frac{\hat{a}_i(b_i)}{(d_i)^2}$  as the weight of each node  $i$  in the network. The pseudo code of simple algorithm 2, denoted by SA<sub>2</sub>, is presented below.

**Table 4-2 SA<sub>2</sub> Algorithm**

```

//Initialization
FOR each node  $i$  {
     $weight = \frac{\hat{a}_i(b_i)}{(d_i)^2}$ ;
}
Add source  $s$  to the  $attack\_tree$ ;

Prim(); //construct the minimum cost spanning tree rooted at s
WHILE ( $total\_attack\_cost < TOTAL\_ATTACK\_BUDGET$  AND there are still
uncompromised nodes) {
    Find node  $i$ , which is uncompromised AND whose  $weight$  is the smallest among
    all uncompromised nodes;
    IF ( $total\_attack\_cost + path\_cost$  of  $i \leq TOTAL\_ATTACK\_BUDGET$ ) {

```

```

    Compromise node  $i$  and all nodes on  $i$ 's path, and add them to the attack_tree;
    total_attack_cost += path_cost of node  $i$ ;
  }
}

```

### 4.1.3 Simple Algorithm 3

We assume that the attacker has complete information about the targeted network in the problem description, and design several heuristics based on this concept. However, it is also important to compare the difference between the performance of an attacker with complete information and an attacker with incomplete information. Thus, in this simple algorithm, we focus on the scenario where the attacker is only aware of the existence of uncompromised nodes through their compromised neighbors.

First, the weight of each node is set to  $\frac{\hat{a}_i(b_i)}{(d_i)^2}$ , as in the two other simple algorithms. After assigning the node weights, we construct an attack tree from the attacker's initial position,  $s$ , by the greedy method. Initially, we create a victim candidate set consisting of nodes directly connected to  $s$ , and include the node with minimal weight in the set of the attack tree. The defense capability of the node should be deducted from attacker's total energy budget. Next, we probe all the neighbors of the node just attacked and add them to the set if they have not been included in the attack tree already. This probing and attacking procedure is repeated until the attacker does not have enough power to compromise another node. The total computational complexity of this heuristic is  $O(|N|^2)$ ; however, it can be reduced to  $O(|N|\log|N|)$  if a heap is used to maintain the victim candidate set. The core of simple algorithm 3, denoted by SA<sub>3</sub>, is described in Table 4-3.



**Table 4-3 SA<sub>3</sub> Algorithm**

```
//Initialization
FOR each node  $i$  {
     $weight = \frac{\hat{a}_i(b_i)}{(d_i)^2}$ ;
}
Add source  $s$  to the attack_tree;

//Construction of Attack Tree
WHILE ( $total\_attack\_cost < TOTAL\_ATTACK\_BUDGET$  AND there are still
uncompromised nodes) {
    Find node  $i$ , whose weight is the smallest among all other nodes' weight in
    victim_candidate_set AND whose defense_capability is less than
    ( $TOTAL\_ATTACK\_BUDGET - total\_attack\_cost$ );
    Compromise node  $i$  and add it to the attack_tree;
     $total\_attack\_cost += defense\_power$  of node  $i$ ;
    Update victim_candidate_set;
}
```

#### 4.1.4 Experiment Environment

The proposed algorithms for the AS model are coded in Visual C++ and run on a PC with an INTEL™ Pentium 4.3GHz CPU. The Iteration Counter Limit and Improve Counter Limit are set to 2000 and 80 respectively; the first-stage relaxation process and the relevant primal algorithm are implemented in iterations 1~1000, and the second-stage relaxation process and the relevant primal algorithm are implemented in iterations 1001~2000. The step size scalar,  $\lambda$ , is initialized as 2 and is halved if the objective function value,  $Z_D$ , does not improve after iterations up to the Improve Counter Limit.

We adopt three kinds of network topology as attack targets. The first type is a grid network, which is a square area composed of  $k \times k$  nodes; the second is a random network, in which each node is connected to several nodes arbitrarily, and the average

degree of each node is set to four, like the grid topologies; and the third is a scale-free network, in which each newly added node connects to two different nodes in the network.

To observe the effect of different information value distribution patterns, we design three kinds of damage distribution mechanisms. The first is random distribution, in which the value of information held by a node is randomly decided; the second is degree-based distribution, in which the higher the degree of a node, the greater the loss that is incurred by an attack; the third is uniform distribution, in which each failed node causes the same amount of damage.

We also design different budget allocation strategies to determine which budget allocation strategy is more effective under different circumstances. The first strategy is uniform budget allocation, whereby each node is allotted the same defense budget; the second is degree-based budget allocation, which allocates the budget according to the percentage of a node's degree over the total degree of the network; the third is damage-based allocation, whereby each node's budget is allocated according to the damage incurred if it is compromised.

As to the function of defense capability,  $\hat{a}_i(b_i)$ , for simplicity, we define it as a linear function. In order to ensure cost-effectiveness, the resulting defense capability must be more than the defense budget invested, or the investment is will not profitable. Here, the cost-benefit ratio is 1:2.

The parameters and scenarios used in our experiments are detailed below.

Table 4-4 Experiment Parameter Settings for the AS Model

<b>Parameters of LR</b>	
<b>Parameters</b>	<b>Value</b>
Iteration Counter Limit	2000
Improve Counter Limit	80
Initial UB	0
Initial Multiplier Value	$\mu_1^0, \mu_2^0, \mu_3^0 = 0,$ $v_1^{1001} = \mu_1^{1000}, v_2^{1001} = \mu_2^{1000},$ $v_3^{1001} = d_m / \hat{a}_m(b_m),$ where $m$ is the critical item and $m \in N$
Initial Scalar of Step Size $\lambda$	2
Test Platform	CPU: INTEL™ Pentium 4.3GHz RAM: 1 GB OS: Microsoft Windows 2000
<b>Parameters of the AS Model</b>	
<b>Parameters</b>	<b>Value</b>
Testing Topology	Grid networks, Random networks, Scale-free networks
Number of Nodes $ N $	49, 100, 400, 900
Total Defense Budget $B$	Equal to Number of Nodes
Total Attack Budget $A$	Equal to Total Defense Budget
Damage Distribution	Random distribution (D <sub>1</sub> ), Degree-based distribution (D <sub>2</sub> ), Uniform distribution (D <sub>3</sub> )
Budget Allocation Strategy	Uniform allocation (B <sub>1</sub> ), Degree-based allocation (B <sub>2</sub> ), Damage-based allocation (B <sub>3</sub> )
Defense Capability $\hat{a}_i(b_i)$	$\hat{a}_i(b_i) = 2b_i + \varepsilon$ , $b_i$ is the budget allocated to node $i$ , $\forall i \in N$

### 4.1.5 Experiment Results

To compare attack behavior under different scenarios, we use the network susceptibility metric to evaluate the degree to which the attacker's objective is achieved. Also, for clarity, solutions to the AS model and simple algorithms are

transformed to the susceptibility of the targeted network after attack; the greater the susceptibility, the more successful the attack. The LR value means the susceptibility calculated by the optimal feasible solution derived by the Lagrangean relaxation process; The LB value is a lower bound on LR obtained from (LR 1) and (LR 2); and SA<sub>1</sub>, SA<sub>2</sub>, and SA<sub>3</sub> are the susceptibilities obtained from simple algorithms 1, 2, and 3 respectively. To evaluate the quality of LR, we calculate the gap between LR and LB by  $\frac{LB-LR}{LR} \times 100\%$ . In addition, the improvement ratio of LR to SA<sub>1</sub>, SA<sub>2</sub>, and SA<sub>3</sub> is calculated by  $\frac{LR-SA_1}{SA_1} \times 100\%$ ,  $\frac{LR-SA_2}{SA_2} \times 100\%$ , and  $\frac{LR-SA_3}{SA_3} \times 100\%$ .



Table 4-5 Experiment Results of Small Networks ( $|N| = 49$ )

Network Topology	Damage Distribution	Budget Allocation	LR (%)	Gap (%)	Improve-	Improve-	Improve-
					ment Ratio to SA <sub>1</sub> (%)	ment Ratio to SA <sub>2</sub> (%)	ment Ratio to SA <sub>3</sub> (%)
Grid Networks	D1	B1	69.41	7.80	2.95	2.51	4.32
		B2	55.45	3.56	0.00	3.16	0.00
		B3	47.92	3.32	0.00	0.00	0.00
	D2	B1	70.31	6.08	5.54	8.79	5.61
		B2	49.27	0.61	0.75	1.50	0.75
		B3	56.25	0.43	1.54	1.54	0.00
	D3	B1	49.65	0.12	0.10	0.50	0.22
		B2	49.27	0.61	0.75	1.50	0.75
		B3	47.92	3.32	0.00	0.00	0.00
Random Networks	D1	B1	72.22	4.97	1.52	4.84	1.18
		B2	67.11	2.34	0.16	3.05	0.00
		B3	47.92	3.32	0.00	0.00	0.00
	D2	B1	72.86	8.77	4.90	5.58	4.35
		B2	49.28	0.82	0.00	0.00	0.00
		B3	60.83	10.00	4.31	9.80	7.41
	D3	B1	49.68	0.08	0.16	0.24	0.07
		B2	49.28	0.82	0.00	0.00	0.00
		B3	47.92	3.32	0.00	0.00	0.00
Scale-free Networks	D1	B1	69.54	4.71	2.06	4.30	2.18
		B2	71.18	1.72	0.00	5.73	0.00
		B3	47.92	3.32	0.00	0.00	0.00
	D2	B1	69.47	18.68	10.53	17.24	5.23
		B2	49.63	0.39	0.65	0.43	0.22
		B3	62.08	20.96	6.49	26.31	3.55
	D3	B1	49.66	0.07	0.07	0.38	0.15
		B2	49.63	0.39	0.65	0.43	0.22
		B3	47.92	3.32	0.00	0.00	0.00

Table 4-5 Experiment Results of Medium-sized Networks ( $|N| = 100$ )

Network Topology	Damage Distribution	Budget Allocation	LR (%)	Gap (%)	Improve-	Improve-	Improve-
					ment Ratio to SA <sub>1</sub> (%)	ment Ratio to SA <sub>2</sub> (%)	ment Ratio to SA <sub>3</sub> (%)
Grid Networks	D1	B1	71.36	4.97	8.98	3.01	5.45
		B2	54.70	0.71	0.00	0.21	0.00
		B3	49.49	0.52	0.00	0.00	0.00
	D2	B1	71.23	5.29	8.99	3.61	6.80
		B2	49.72	0.12	0.23	0.23	0.23
		B3	54.55	0.39	0.00	0.00	0.00
	D3	B1	49.82	0.06	0.06	0.05	0.03
		B2	49.72	0.12	0.23	0.23	0.23
		B3	49.49	0.52	0.00	0.00	0.00
Random Networks	D1	B1	73.32	3.25	1.76	4.28	2.57
		B2	67.80	0.38	0.00	3.03	0.07
		B3	49.49	0.52	0.00	0.00	0.00
	D2	B1	74.56	9.58	4.84	6.45	7.77
		B2	49.70	0.28	0.00	0.00	0.00
		B3	61.21	9.99	3.43	8.24	4.16
	D3	B1	49.84	0.03	0.04	0.10	0.03
		B2	49.70	0.28	0.00	0.00	0.00
		B3	49.49	0.52	0.00	0.00	0.00
Scale-free Networks	D1	B1	71.18	2.75	0.53	7.10	2.70
		B2	74.14	0.21	0.00	6.56	0.00
		B3	49.49	0.52	0.00	0.00	0.00
	D2	B1	72.07	16.01	9.44	19.46	5.08
		B2	49.77	0.25	0.10	0.21	0.10
		B3	63.84	19.25	6.03	23.12	0.65
	D3	B1	49.84	0.03	0.03	0.07	0.05
		B2	49.77	0.25	0.10	0.21	0.10
		B3	49.49	0.52	0.00	0.00	0.00

Table 4-6 Experiment Results of Large Networks ( $|N| = 400$ )

Network Topology	Damage Distribution	Budget Allocation	LR (%)	Gap (%)	Improve-	Improve-	Improve-
					ment Ratio to SA <sub>1</sub> (%)	ment Ratio to SA <sub>2</sub> (%)	ment Ratio to SA <sub>3</sub> (%)
Grid Networks	D1	B1	71.72	4.90	8.45	1.71	5.81
		B2	52.47	0.15	0.00	0.00	0.00
		B3	49.87	0.13	0.00	0.00	0.00
	D2	B1	71.45	5.74	8.86	0.77	5.79
		B2	49.93	0.03	0.11	0.11	0.11
		B3	52.38	0.08	0.00	0.00	0.00
	D3	B1	49.96	0.01	0.01	0.01	0.01
		B2	49.93	0.03	0.11	0.11	0.11
		B3	49.87	0.13	0.00	0.00	0.00
Random Networks	D1	B1	72.30	3.08	1.57	4.10	1.52
		B2	68.22	0.09	0.00	3.41	0.02
		B3	49.87	0.13	0.00	0.00	0.00
	D2	B1	71.65	10.23	8.26	4.30	2.59
		B2	49.91	0.11	0.00	0.00	0.00
		B3	61.00	10.28	2.20	5.56	3.95
	D3	B1	49.96	0.01	0.00	0.01	0.00
		B2	49.91	0.11	0.00	0.00	0.00
		B3	49.87	0.13	0.00	0.00	0.00
Scale-free Networks	D1	B1	72.61	2.53	1.07	6.00	2.22
		B2	74.59	0.05	0.00	8.12	0.00
		B3	49.87	0.14	0.00	0.00	0.00
	D2	B1	71.61	21.34	9.65	16.99	0.22
		B2	49.95	0.05	0.03	0.05	0.05
		B3	64.11	22.70	2.17	34.34	0.08
	D3	B1	49.96	0.01	0.00	0.01	0.00
		B2	49.95	0.05	0.03	0.05	0.05
		B3	49.87	0.14	0.00	0.00	0.00

Table 4-7 Experiment Results of Extra-large Networks ( $|N| = 900$ )

Network Topology	Damage Distribution	Budget Allocation	LR (%)	Gap (%)	Improve-	Improve-	Improve-
					ment Ratio to SA <sub>1</sub> (%)	ment Ratio to SA <sub>2</sub> (%)	ment Ratio to SA <sub>3</sub> (%)
Grid Networks	D1	B1	71.94	3.76	8.61	0.91	4.07
		B2	51.66	0.07	0.00	0.00	0.00
		B3	49.94	0.09	0.00	0.00	0.00
	D2	B1	71.81	4.23	8.98	1.11	3.77
		B2	49.97	0.01	0.05	0.05	0.05
		B3	51.61	0.04	0.00	0.00	0.00
	D3	B1	49.98	0.01	0.01	0.00	0.00
		B2	49.97	0.01	0.05	0.05	0.05
		B3	49.94	0.09	0.00	0.00	0.00
Random Networks	D1	B1	72.88	3.08	0.18	3.84	2.09
		B2	69.12	0.05	0.02	3.25	0.01
		B3	49.94	0.06	0.00	0.00	0.00
	D2	B1	72.18	10.64	4.86	4.29	3.13
		B2	49.97	0.03	0.00	0.00	0.00
		B3	60.85	11.88	2.02	4.63	3.21
	D3	B1	49.98	0.01	0.00	0.00	0.00
		B2	49.97	0.03	0.00	0.00	0.00
		B3	49.94	0.06	0.00	0.00	0.00
Scale-free Networks	D1	B1	72.64	2.79	0.98	5.41	2.27
		B2	74.71	0.02	0.00	8.02	0.00
		B3	49.94	0.31	0.00	0.00	0.00
	D2	B1	72.26	20.25	1.62	24.82	1.27
		B2	49.98	0.02	0.00	0.03	0.02
		B3	64.29	23.96	3.89	36.48	0.24
	D3	B1	49.98	0.01	0.00	0.00	0.00
		B2	49.98	0.02	0.00	0.03	0.02
		B3	49.94	0.31	0.00	0.00	0.00



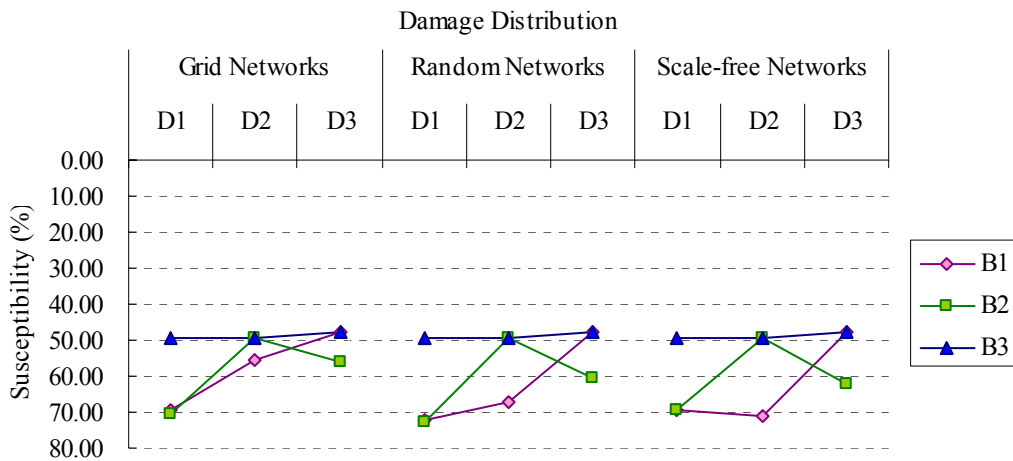


Figure 4-1 Susceptibility of Small Networks under Different Scenarios ( $|N| = 49$ )

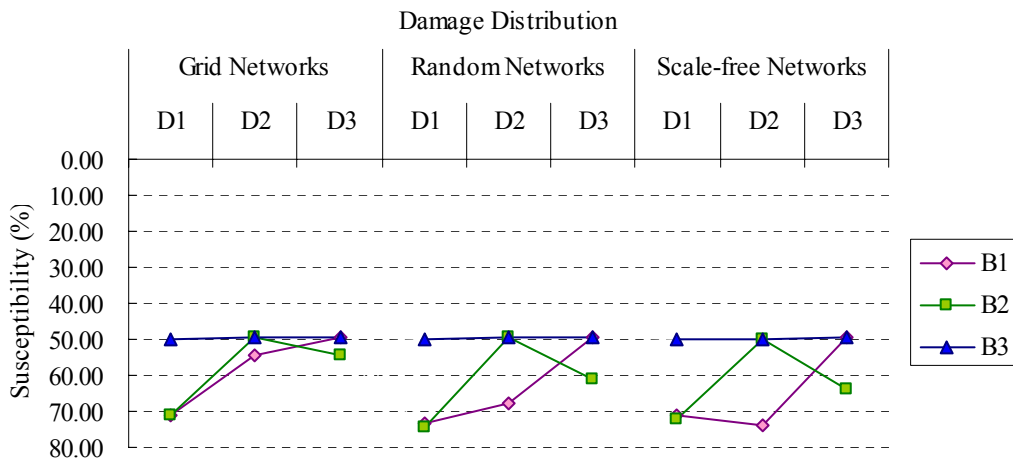


Figure 4-2 Susceptibility of Medium-sized Networks under Different Scenarios ( $|N| = 100$ )

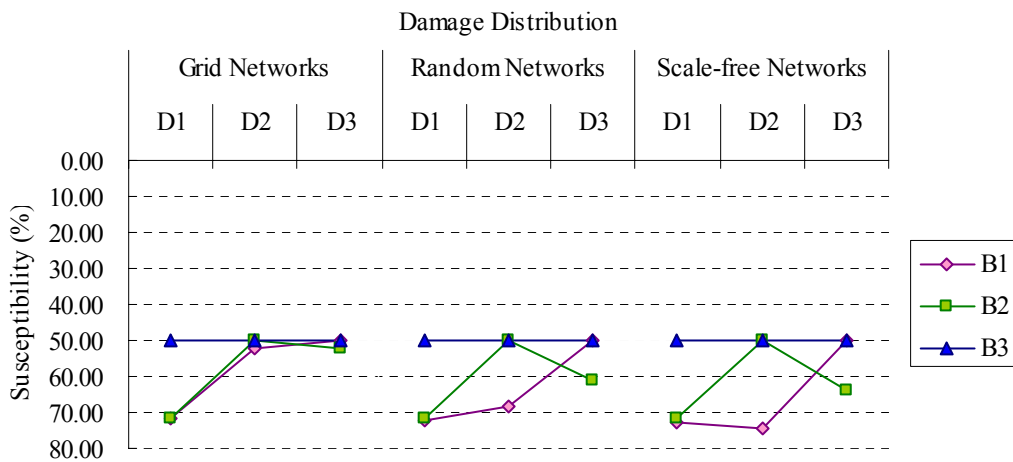


Figure 4-3 Susceptibility of Large Networks under Different Scenarios ( $|N| = 400$ )

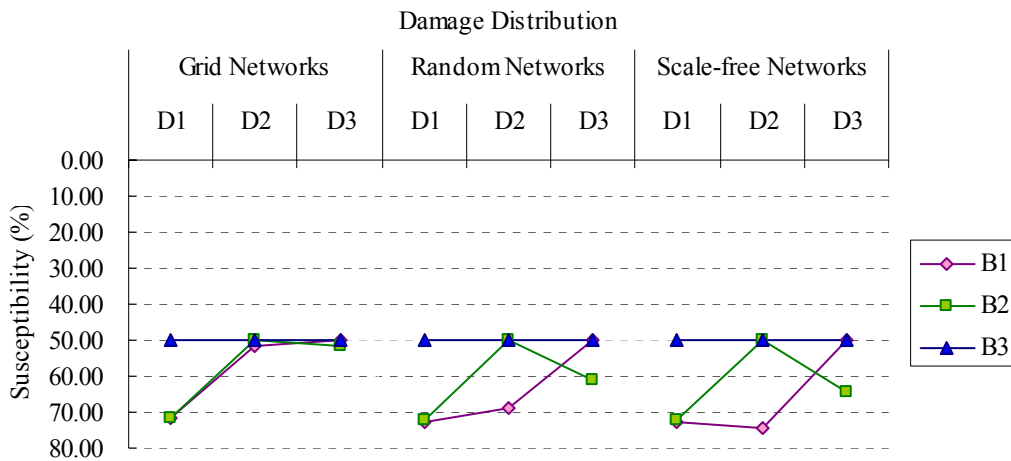


Figure 4-4 Susceptibility of Extra-large Networks under Different Scenarios ( $|N| = 900$ )

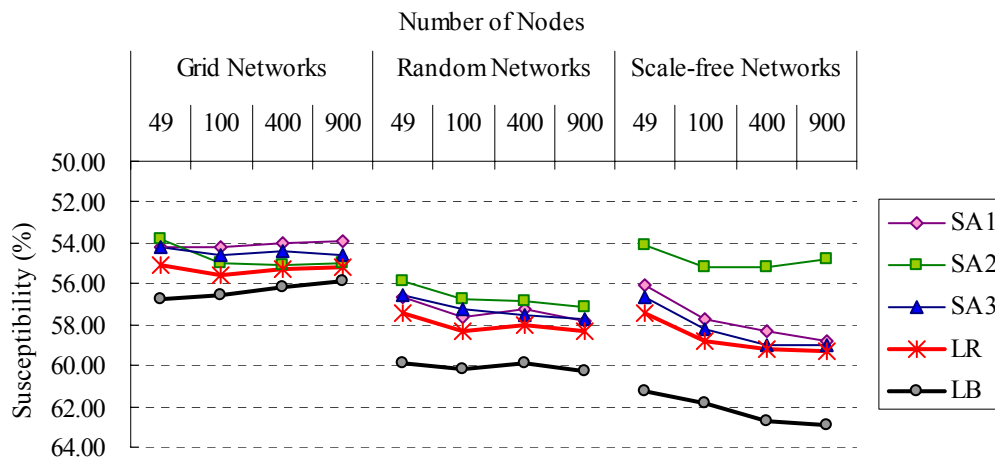


Figure 4-5 Susceptibility of Different Network Sizes and Topologies

## 4.1.6 Discussion of Results

Figures 4-1 to 4-4 show the susceptibility of the targeted network under different topology types, numbers of nodes, and damage distribution patterns. From these figures, we observe:

- Networks with budget allocation strategy B3 are the most robust and therefore the most difficult for an attacker to compromise. This finding is consistent with the common idea that defense resources should be allocated according to the importance of each node. According to this result, the

budget allocation strategy B2 under damage distribution D2 and B3 under D2 can achieve the same effect as B3.

- For grid networks, the network susceptibility of the B1 and B2 strategies is close, and the gap between them decreases with the growth of the networks. This is due to one of the main features of grid networks – the degree of each node is four, except for nodes on edges. The larger the network size, the more the average degree of a grid network will approach four. Thus, the difference between B1 and B2 disappears as the network grows and most nodes in the network have the same degree.
- Under the scenarios of the D1 allocation pattern, the B1 and B2 strategies make the targeted network highly susceptible to attack. Because the information value of each node is decided randomly, the two divergent strategies can not protect important nodes effectively.
- Networks under the D3 scenario have the lowest susceptibility among the three damage distribution patterns. Although wrong defense budget allocation strategies still cause high network susceptibility, generally speaking, the network susceptibility of the D3 pattern is lower than that of the D1 and D2 patterns. This result indicates that a network is more robust if “all nodes are created equal,” because the attacker can not target nodes selectively, and the number of nodes that are compromised directly decides the maximum total profit.
- The degree distribution of a network’s topology affects the network’s susceptibility. Take the B1 strategy under the D2 distribution pattern for example. In this scenario, the susceptibility of grid networks is the lowest,

and that of scale-free networks is the highest among the three network topologies. Due to the uniform degree of each node in a grid network, the results of B1 and B2 are similar; however the node degrees in random networks are irregular, so the strategy of treating each node equally fails to reflect the discrepancy between the nodes. The situation is more serious in scale-free networks because the power-law degree distribution induces tremendous divergence between the average degree and the actual degree of each node.

Figure 4-5 compares the solution quality of the proposed Lagrangean relaxation-based algorithm with simple algorithms 1, 2, and 3, and demonstrates the gap between LRs and LBs. The value of each point on the figure is the average susceptibility of different damage distribution patterns and different defense budget allocation strategies under same network size and topology. From the figure, we observe several trends.

- Our proposed heuristic outperforms the three simple algorithms in all cases. From Figure 4-5, we observe that, while the three simple algorithms perform well in some network topologies, our attack strategy always causes the highest network susceptibility in all three topologies. This indicates that our proposed Lagrangean relaxation-based algorithm is not only capable of solving the AS model, but it is also applicable to various types of network topology. The gaps between LRs and LBs are small, which shows that our proposed approach can derive a near-optimal solution to the AS model.
- Simple algorithm 2 performs very well in grid networks, but fails in scale-free networks. The main property of the algorithm is that the attacker decides the target first, and then finds an attack path to reach that node. This

strategy is useful when there are multiple paths between the source and the target because the attacker can make a detour when encountering nodes with high defense capability. However, in the case of scale-free networks, the connectivity between nodes is maintained by a few hubs, that is, very few paths exist between the source and the target. Therefore, the attacker can not avoid the nodes with the highest cost when constructing an attack path to the target node, so the attack budget is consumed rapidly.

- Simple algorithm 3 performs reasonably well in all types of network, especially scale-free networks. Theoretically, the solution quality of this algorithm should be worse than that of the other algorithms because of its local-information-awareness property. However, the results show that there is only a small gap between attack strategies with complete information and those with local information. One possible reason is that when an attacker has too much information, he may not be able to utilize fully to develop the perfect attack strategy. On the other hand, attack strategies based on local information can generate almost the same susceptibility as that caused by strategies with complete information in scale-free networks. This is because the “six degrees of separation” property holds in scale-free networks, and an attacker can collect complete information about the targeted network once he has compromised several hub nodes.
- Generally, scale-free networks are more susceptible than the other two topologies; grid networks are the least susceptible. This phenomenon results from the effects of the B1 and B2 strategies, since the susceptibility of all networks is the same under the B3 strategy. Moreover, it is also consistent with the findings of previous research that scale-free networks are more

vulnerable to malicious attacks. As most nodes in scale-free networks are connected to just a few hubs, the number of directly reachable target nodes increases enormously once the hubs have been taken. In contrast, the regular structure of a grid network makes it difficult for an attacker to reach valuable nodes arbitrarily, so grid networks are less susceptible to information theft.

## 4.2 Computational Experiments with the DRAS Model

### 4.2.1 Experiment Environment

The proposed algorithms for the DRAS model are coded in Visual C++ and run on a PC with an INTEL™ Pentium 4.3GHz CPU. The Iteration Counter Limit and Improve Counter Limit are set to 500 and 20 respectively. The step size scalar,  $\theta$ , is initialized as 0.5 and is halved if the objective function value,  $Z_{IP,1}$ , does not improve after the iterations up to the Improve Counter Limit.

In the DRAS model, the attacker tries to steal as much information as possible under a certain defense budget allocation strategy. Thus, an initial budget allocation strategy must be provided before the first attack. From the results of the AS model, we conclude that the B3 allocation strategy is the best of the three given strategies. Therefore, the B3 strategy is adopted as the initial defense resource allocation strategy for the DRAS model.

After each attack, the defender adjusts each node's allotted budget according to the budget reallocation strategies. Here, three reallocation strategies are chosen to adjust each node's budget. They are the same as the defense budget allocation strategies in the AS model.

**Table 4-8 Experiment Parameter Settings for the DRAS Model**

<b>Parameters of Adjusment_Procedure</b>	
<b>Parameters</b>	<b>Value</b>
Iteration Counter Limit	500
Improve Counter Limit	20
Initial Scalar of Step Size $\theta$	0.5
Test Platform	CPU: INTEL™ Pentium 4.3GHz RAM: 1 GB OS: Microsoft Windows 2000
<b>Parameters of the DRAS Model</b>	
<b>Parameters</b>	<b>Value</b>
Testing Topology	Grid networks, Random networks, Scale-free networks
Number of Nodes $ N $	25, 49, 100
Total Defense Budget $B$	Equal to Number of Nodes
Total Attack Budget $A$	Equal to Total Defense Budget
Damage Distribution	Random distribution ( $D_1$ ), Degree-based distribution ( $D_2$ ), Uniform distribution ( $D_3$ )
Initial Budget Allocation Strategy	Damage-based allocation ( $B_3$ )
Budget Reallocation Strategy	Uniform allocation ( $B_1$ ), Degree-based allocation ( $B_2$ ), Damage-based allocation ( $B_3$ )
Defense Capability $\hat{a}_i(b_i)$	$\hat{a}_i(b_i) = 2b_i + \varepsilon$ , $b_i$ is the budget allocated to node $i$ , $\forall i \in N$

## 4.2.2 Experiment Results

In the experiments, we use the survivability of the targeted network, which is determined by the equilibrium of the offense-defense scenario, to evaluate the performance of different defense resource reallocation strategies. Solutions to the DRAS model are transformed to the equilibrium survivability of the targeted network; the higher the survivability, the better the reallocation strategy. The Init. Surv. value represents the network survivability under the initial defense budget allocation strategy, and the value of Opt. Surv. is the equilibrium of the network's survivability

resulting from the budget reallocation strategy. The improvement ratio of Opt. Surv. to

Init. Surv. is calculated by  $\frac{\text{Opt. Surv.}-\text{Init. Surv.}}{\text{Init. Surv.}} \times 100\%$ .

**Table 4-9 Experiment Results of Extra-small Networks ( $|N| = 25$ )**

<b>Network Topology</b>	<b>Damage Distribution</b>	<b>Init. Surv. (%)</b>	<b>Budget Allocation</b>	<b>Opt. Surv. (%)</b>	<b>Imp. Ratio of Opt. Surv. (%)</b>
<b>Grid Networks</b>	<b>D1</b>	50.75	<b>B1</b>	51.15	0.80
			<b>B2</b>	52.49	2.16
			<b>B3</b>	54.17	0.00
	<b>D2</b>	51.44	<b>B1</b>	50.95	0.40
			<b>B2</b>	52.76	2.70
			<b>B3</b>	54.17	0.00
	<b>D3</b>	54.17	<b>B1</b>	51.70	1.93
			<b>B2</b>	52.76	2.70
			<b>B3</b>	54.17	0.00
<b>Random Networks</b>	<b>D1</b>	50.71	<b>B1</b>	51.42	1.45
			<b>B2</b>	52.16	1.69
			<b>B3</b>	54.17	0.00
	<b>D2</b>	51.33	<b>B1</b>	51.07	0.74
			<b>B2</b>	52.56	2.51
			<b>B3</b>	54.17	0.00
	<b>D3</b>	54.17	<b>B1</b>	52.17	2.96
			<b>B2</b>	52.56	2.51
			<b>B3</b>	54.17	0.00
<b>Scale-free Networks</b>	<b>D1</b>	50.72	<b>B1</b>	51.79	2.16
			<b>B2</b>	53.25	4.58
			<b>B3</b>	54.17	0.00
	<b>D2</b>	51.00	<b>B1</b>	50.96	0.47
			<b>B2</b>	55.03	8.22
			<b>B3</b>	54.17	0.00
	<b>D3</b>	54.17	<b>B1</b>	51.61	1.80
			<b>B2</b>	55.03	8.22
			<b>B3</b>	54.17	0.00



**Table 4-10 Experiment Results of Small Networks ( $|N| = 49$ )**

<b>Network Topology</b>	<b>Damage Distribution</b>	<b>Init. Surv. (%)</b>	<b>Budget Allocation</b>	<b>Opt. Surv. (%)</b>	<b>Imp. Ratio of Opt. Surv. (%)</b>
<b>Grid Networks</b>	<b>D1</b>	50.36	<b>B1</b>	50.49	0.27
			<b>B2</b>	51.03	0.49
			<b>B3</b>	52.08	0.00
	<b>D2</b>	50.79	<b>B1</b>	50.50	0.28
			<b>B2</b>	51.03	0.49
			<b>B3</b>	52.08	0.00
	<b>D3</b>	52.08	<b>B1</b>	50.69	0.67
			<b>B2</b>	51.03	0.49
			<b>B3</b>	52.08	0.00
<b>Random Networks</b>	<b>D1</b>	50.33	<b>B1</b>	50.66	0.67
			<b>B2</b>	51.63	1.85
			<b>B3</b>	52.08	0.00
	<b>D2</b>	50.72	<b>B1</b>	50.48	0.31
			<b>B2</b>	51.50	1.58
			<b>B3</b>	52.08	0.00
	<b>D3</b>	52.08	<b>B1</b>	51.37	2.10
			<b>B2</b>	51.50	1.58
			<b>B3</b>	52.08	0.00
<b>Scale-free Networks</b>	<b>D1</b>	50.34	<b>B1</b>	50.51	0.35
			<b>B2</b>	51.41	2.04
			<b>B3</b>	52.08	0.00
	<b>D2</b>	50.36	<b>B1</b>	50.94	1.20
			<b>B2</b>	52.25	3.81
			<b>B3</b>	52.08	0.00
	<b>D3</b>	52.08	<b>B1</b>	50.72	0.77
			<b>B2</b>	52.25	3.81
			<b>B3</b>	52.08	0.00

**Table 4-11 Experiment Results of Medium-sized Networks ( $|N| = 100$ )**

<b>Network Topology</b>	<b>Damage Distribution</b>	<b>Init. Surv. (%)</b>	<b>Budget Allocation</b>	<b>Opt. Surv. (%)</b>	<b>Imp. Ratio of Opt. Surv. (%)</b>
<b>Grid Networks</b>	<b>D1</b>	50.17	<b>B1</b>	50.18	0.01
			<b>B2</b>	50.42	0.11
			<b>B3</b>	50.51	0.00
	<b>D2</b>	50.36	<b>B1</b>	50.17	0.00
			<b>B2</b>	50.48	0.23
			<b>B3</b>	50.51	0.00
	<b>D3</b>	50.51	<b>B1</b>	50.27	0.19
			<b>B2</b>	50.48	0.23
			<b>B3</b>	50.51	0.00
<b>Random Networks</b>	<b>D1</b>	50.16	<b>B1</b>	50.54	0.76
			<b>B2</b>	50.67	0.70
			<b>B3</b>	50.51	0.00
	<b>D2</b>	50.32	<b>B1</b>	50.16	0.00
			<b>B2</b>	51.22	1.81
			<b>B3</b>	50.51	0.00
	<b>D3</b>	50.51	<b>B1</b>	51.03	1.76
			<b>B2</b>	51.22	1.81
			<b>B3</b>	50.51	0.00
<b>Scale-free Networks</b>	<b>D1</b>	50.16	<b>B1</b>	50.16	0.00
			<b>B2</b>	50.61	0.77
			<b>B3</b>	50.51	0.00
	<b>D2</b>	50.22	<b>B1</b>	50.47	0.62
			<b>B2</b>	52.17	3.90
			<b>B3</b>	50.51	0.00
	<b>D3</b>	50.51	<b>B1</b>	50.45	0.59
			<b>B2</b>	52.17	3.90
			<b>B3</b>	50.51	0.00

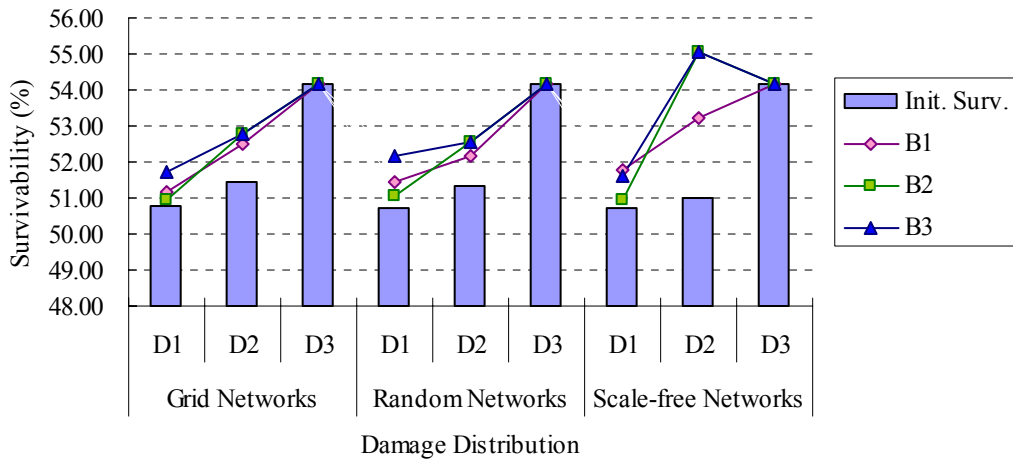


Figure 4-6 Survivability of Extra-small Networks under Different Scenarios ( $|N| = 25$ )

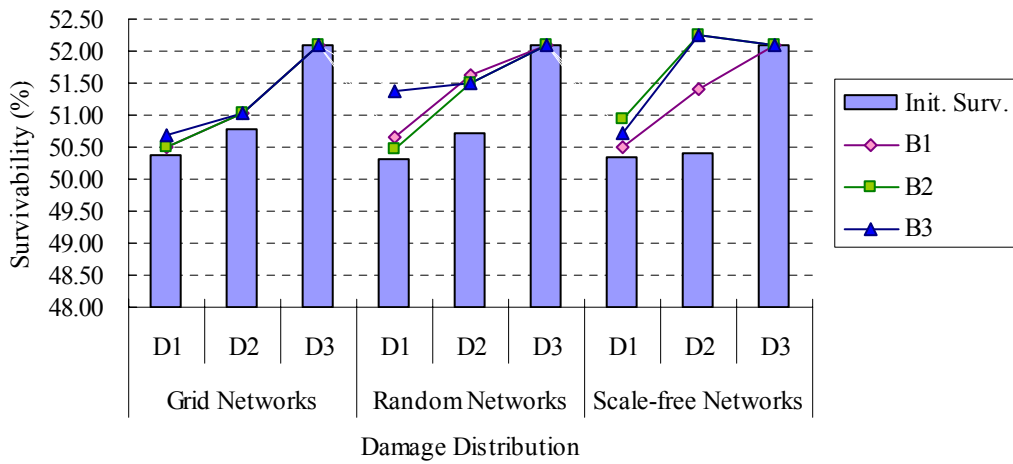


Figure 4-7 Survivability of Small Networks under Different Scenarios ( $|N| = 49$ )

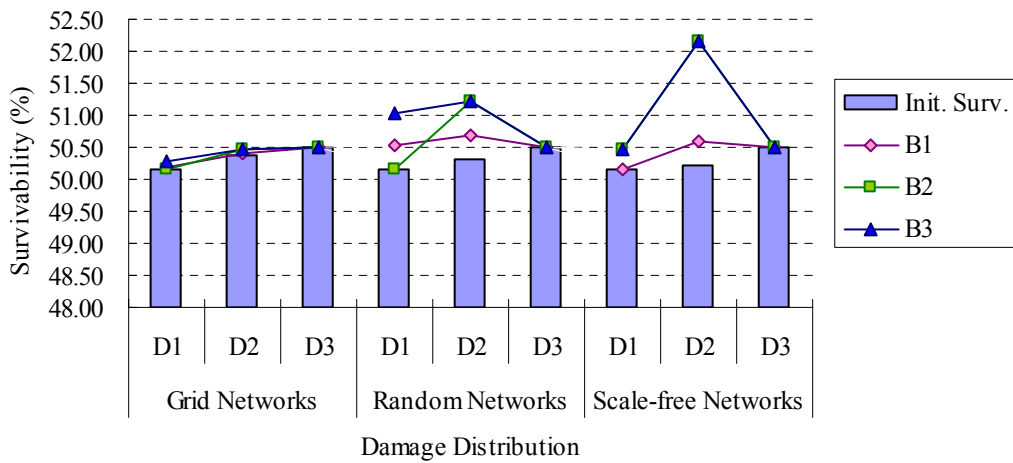


Figure 4-8 Survivability of Medium-sized Networks under Different Scenarios ( $|N| = 100$ )

### 4.2.3 Discussion of Results

Figures 4-6 ~ 4-8 display the equilibrium survivability of the targeted networks under different topology types, numbers of nodes, and damage distribution patterns.

From the figures, we can make several observations:

- “The rich get richer, and the poor get poorer” is the best reallocation strategy for scale-free networks under the D2 distribution. For scale-free networks, the initial survivability of the D2 scenario under the B3 strategy is low. However, by applying the B3 strategy repeatedly, the equilibrium survivability of the D2 scenario outperforms that of the other scenarios; moreover, this case becomes the most robust among all the experimental scenarios. Since the importance of nodes depends on their degree, the result confirms the findings of previous research [14][16] that the protection of hubs in scale-free networks must be enhanced.
- The survivability of networks under the D3 distribution can not be improved by defense budget adjustment procedure. As noted in Section 4.1.6, the initial network survivability of the D3 scenario is the highest among three damage distribution patterns; however, no improvement is made after the resource adjustment procedure. One possible reason is that the nodes are equal in importance; thus, the key-node-oriented reallocation of the defense budget is meaningless.
- In most scenarios, the B3 defense resource reallocation strategy can enhance network survivability more than the B1 and B2 strategies. This phenomenon is most obvious in random networks under D1 distribution. Once again, “the rich get richer, and the poor get poorer” proves to be the

best defense budget allocation strategy, since we allocate defense resources according to the value of the information held by each node initially, and then reallocate the resources repeatedly.

- In Section 4.1.6, we observe that grid networks are the most robust among three testing topologies generally. However, after applying our proposed defense resource reallocation strategy, the equilibrium survivability of random networks and scale-free networks transcend that of grid networks. This implies that random networks and scale-free can be very robust as long as right defense strategies are applied.



## Chapter 5 Conclusion and Future Work

### 5.1 Conclusion

The ubiquitous nature of the Internet has made it a nest of cyber-crimes, which render the concept of “completely secure systems and networks” obsolete, and incur inestimable damage and loss to victims. Information theft is one of the most damaging cyber-crimes, yet it is easily missed because its attack behavior does not alert victims, but makes them unwitting accomplices instead. Thus, network operators not only need to protect their networks against information theft, but must also prevent their networks from being used as hop-sites.

In this thesis we have addressed the attack-defense scenario in terms of information theft, where an attacker attempts to steal information from a targeted network and maximize his gained profit, while the operator of the network tries to minimize the impact of attacks through a proper defense resource allocation strategy. Both the attack strategy and the defense resource allocation strategy must be adjusted repeatedly to maintain equilibrium.

The key contribution of this research is the development of mathematical models of AS and DRAS. We successfully model the interaction between attackers and defenders in the real world into well-formulated mathematical models, which are then solved by the proposed heuristics. This is a breakthrough in the topic of network attacks since previous research seldom modeled real-world attack behavior in this

way [15]. Through mathematical forms, we can induce generic results and apply them to similar real-world scenarios that were only addressed by individual case studies in the past.

The novel network survivability and susceptibility metrics represent another contribution of this thesis. In order to evaluate the performance of different attack strategies and defense resource allocation strategies, we have proposed two complementary metrics: susceptibility and survivability. The metrics reflect the amount of profit gained by an attacker, so that both the attacker and the defender can gauge the survivability of the targeted network, and can adjust their strategies accordingly.

We have also studied several different network topologies and observed their susceptibility against information theft under different defense resource allocation strategies. We then adjusted the defense strategies to improve their survivability. The experiment results show that grid networks are the least susceptible to information theft, while scale-free network are the most susceptible. However, through a proper defense resource allocation strategy, the differences in survivability of different topologies can be reduced. Most importantly, we have developed an engineering guideline for the network defender. Its states that the best defense resource allocation strategy is the one based on the concept: “the rich get richer, and the poor get poorer.”

## 5.2 Future Work

In the following, we highlight several issues and concepts that could be studied further.

- **Function of Defense Capability**

In this research, we adopt a linear defense capability function,  $\hat{a}_i(b_i) = 2b_i + \varepsilon$ , in the computational experiments. However, a concave function is more reasonable when addressing the relation between the defense budget and the defense capability. According to the “Law of Diminishing Marginal Utility”, the marginal benefit, i.e., the additional defense capability derived from an additional unit of defense budget, declines as the defense budget increases. Thus, concave functions, e.g. log functions, may describe the real situation more accurately.

- **Discussion of Special Cases**

During the computational experiment phase, we observed several abnormal results in the DRAS model. These results indicate that there may be better defense resource allocation strategies when a few “choke points” exist in a network. The survivability of this kind of network improves substantially if the choke points are well-defended, and exceeds the average survivability of networks without choke points. This is because the choke points are the gates to other nodes in the network, and the other nodes can not be compromised unless the choke points have been taken. Thus, reinforced defense of these nodes would not only stop the attacker, but would also consume a huge amount of the attacker’s budget. However, how identifying the most important choke points of a network is still a



challenging issue and we hope to study this area thoroughly in the future.

- **Secret Sharing Scheme Concept**

In our problem description, we assumed that once a node has been compromised, the attacker can get all the valuable information held by that node. However, in network security research, the concept of a “secret sharing scheme” is often used. Under the scheme, each node contains a fragment of important and sensitive information, which is useless unless all the fragments of information about the secret can be retrieved. Therefore, several nodes form a group that keep a secret, which only be stolen if all members of the group are compromised.

Extending the DRAS model, we can model the concept of the secret sharing scheme as the following formulation, denoted by (IP 3). Most of the notations used in the formulation are the same as those in the DRAS model; extra notations are listed in Table 5-1.

**Table 5-1 Extra Notations Used in (IP 3)**

<b>Given Parameters</b>	
<b>Notion</b>	<b>Description</b>
$G$	The index set of all sensitive information groups in the network
$s_g$	Damage incurred by compromising all members of group $g$ , where $g \in G$
$\sigma_{gi}$	An indicator function, which is 1 if node $i$ is in sensitive information group $g$ , and 0 otherwise (where $i \in N, g \in G$ )
<b>Decision Variable</b>	
<b>Notion</b>	<b>Description</b>
$z_g$	1 if all members of group $g$ are compromised, and 0 otherwise (where $g \in G$ )

**Objective function:**

$$Z_{IP3} = \min_{b_i} \max_{y_i, a_i, z_g} \sum_{i \in N} d_i y_i + \sum_{g \in G} s_g z_g \quad (\text{IP 3})$$

**Subject to:**

$$\sum_{w \in W} \sum_{p \in P_w} x_p \delta_{pi} \leq (|N| - 1) y_i \quad \forall i \in N \quad (\text{IP 3.1})$$

$$\sum_{p \in P_w} x_p = y_i \quad \forall i \in N, w = (s, i) \quad (\text{IP 3.2})$$

$$\sum_{p \in P_w} x_p \leq 1 \quad \forall w \in W \quad (\text{IP 3.3})$$

$$x_p = 0 \text{ or } 1 \quad \forall p \in P_w, w \in W \quad (\text{IP 3.4})$$

$$y_i = 0 \text{ or } 1 \quad \forall i \in N \quad (\text{IP 3.5})$$

$$0 \leq b_i \leq B \quad \forall i \in N \quad (\text{IP 3.6})$$

$$\sum_{i \in N} b_i \leq B \quad (\text{IP 3.7})$$

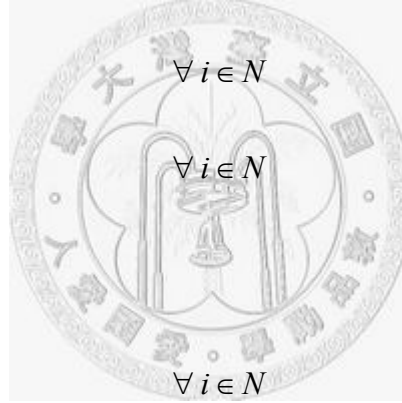
$$0 \leq a_i \leq \hat{a}_i(b_i) \quad \forall i \in N \quad (\text{IP 3.8})$$

$$\sum_{i \in N} a_i \leq A \quad (\text{IP 3.9})$$

$$\hat{a}_i(b_i) y_i \leq a_i \quad \forall i \in N \quad (\text{IP 3.10})$$

$$z_g = 0 \text{ or } 1 \quad \forall g \in G \quad (\text{IP 3.11})$$

$$\sum_{i \in N} \sigma_{gi} z_g \leq \sum_{i \in N} \sigma_{gi} y_i \quad \forall g \in G. \quad (\text{IP 3.12})$$



### Explanation of the mathematical formulation:

- Objective function: The objective is to minimize the maximized total damage incurred by compromising single nodes,  $\sum_{i \in N} d_i y_i$ , and extra damage incurred by compromising all members in some sensitive information group,  $\sum_{g \in G} s_g z_g$ . In the inner problem, an attacker tries to maximize the damage to the targeted network by deciding which nodes or groups to attack, i.e., the  $y_i$  value of each node  $i$  and the  $z_g$  value of each group  $g$ . In the outer problem, the defender tries to minimize the damage caused by the attacker by allocating the defense resources,  $b_i$ , to each node appropriately.
- Constraints (IP 3.1) ~ (IP 3.10) are the same as Constraints (IP 1.1) ~ (IP 1.10) in the DRAS model.
- Constraints (IP 3.11) and (IP 3.12) state that a sensitive group  $g$  can only be compromised if all members of the group have been taken by the attacker.

In this research, we have modeled real-world offense-defense scenarios of information leakage/theft. None the less, the future research issues mentioned above have the potential to substantially improve the accuracy and practicability of our models. Thus, follow-up research will be conducted, and more supplements will be added to enhance our models in the future.

## References

- [1] L.A. Gordon, M.P. Loeb, W. Lucyshyn, and R. Richardson, “2005 CSI/FBI Computer Crime and Security Survey”, *Computer Security Institute*, 2005, <http://GoCSI.com>.
- [2] L.A. Gordon, M.P. Loeb, W. Lucyshyn, and R. Richardson, “2006 CSI/FBI Computer Crime and Security Survey”, *Computer Security Institute*, 2006, <http://GoCSI.com>.
- [3] J.C. Knight and K.J. Sullivan, “On the Definition of Survivability,” *Technical Report CS-TR-33-00, Department of Computer Science, University of Virginia*, December 2000.
- [4] J.C. Knight, E.A. Strunk, and K.J. Sullivan, “Towards a Rigorous Definition of Information System Survivability,” *Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX 2003)*, Volume 1, pp.78-89, April 2003.
- [5] Y. Liu and K.S. Trivedi, “A General Framework for Network Survivability Quantification,” *Proceedings of the 12<sup>th</sup> GI/ITG Conference on Measuring, Modeling and Evaluation of Computer and Communication Systems*, September 2004.
- [6] R.J. Ellison, D.A. Fisher, R.C. Linger, H.F. Lipson, T.A. Longstaff, and N.R. Mead, “Survivable Network Systems: An Emerging Discipline,” *Technical Report CMU/SEI-97-TR-013, Software Engineering Institute, Carnegie Mellon University*, November 1997 (Revised: May 1999).
- [7] P. Tarvainen, “Survey of the Survivability of IT Systems,” *The 9<sup>th</sup> Nordic*

*Workshop on Secure IT-systems*, November 2004.

- [8] “Technical Report on Enhanced Network Survivability Performance,” T1A1.2 Working Group on Network Survivability Performance, February 2001.
- [9] V.R. Westmark, “A Definition for Information System Survivability,” *Proceedings of the 37<sup>th</sup> IEEE Hawaii International Conference on System Sciences*, Volume 9, p. 90303.1, 2004.
- [10] M.S. Deutsch and R.R. Willis, *Software Quality Engineering: A Total Technical and Management Approach*, Englewood Cliffs, NJ: Prentice-Hall, 1988.
- [11] S.C. Liew and K.W. Lu, “A Framework for Network Survivability Characterization,” *IEEE Journal on Selected Areas in Communications*, Volume 12, No. 1, pp. 52-58, January 1994 (ICC, 1992).
- [12] S. Louca, A. Pitsillides and G. Samaras, “On Network Survivability Algorithms Based on Trellis Graph Transformations,” *Fourth IEEE Symposium on Computers and Communications (ISCC’99)*, pp. 235-243, July 1999,
- [13] “Telecom Glossary 2000 (American National Standard, T1.523-2001),” Alliance for Telecommunications Industry Solutions, <http://www.atis.org/tg2k/>.
- [14] R. Albert, H. Jeong, and A.-L. Barabási, “Error and Attack Tolerance of Complex Networks,” *Nature*, Volume 406, pp. 378-382, July 2000.
- [15] A. Stewart, “On Risk: Perception and Direction,” *Computers and Security*, Volume 23, pp. 362-370, May 2004.
- [16] S.-T. Park, A. Khrabrov, D.M. Pennock, S. Lawrence, C.L. Giles, and L.H. Ungar, “Static and Dynamic Analysis of the Internet’s Susceptibility to Faults and Attacks,” *Proceeding of the 22<sup>nd</sup> Annual Conference of the IEEE Computer*

*and Communications Societies, 2003.*

- [17] M. Keshtgary, F.A. Al-Zahrani, and A.P. Jayasumana, "Network Survivability Performance Evaluation with Applications in WDM Networks with Wavelength Conversion," *Proceedings of the 29<sup>th</sup> Annual IEEE International Conference on Local Computer Networks*, 2004.
- [18] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On Power-Law Relationships of the Internet Topology," *ACM SIGCOIMM Computer Communications Review*, Volume 29, Number 4, pp. 251-263, September 1999.
- [19] G. Siganos, M. Faloutsos, P. Faloutsos, and C. Faloutsos, "Power-Laws and the AS-level Internet Topology," *IEEE/ACM Transactions on Networking*, Volume 11, Issue 4, pp. 514-524, 2003.
- [20] A.-L. Barabási and R. Albert, "Emergence of Scaling in Random Networks," *Science*, Volume 286, pp. 509-512, October 1999.
- [21] P. Magadevan, D. Krioukov, M. Fomenkov, B. Huffaker, X. Dimitropoulos, K. Claffy, and A. Vahdat, "The Internet AS-level Topology: Three Data Sources and One Definitive Metric," *ACM SIGCOMM Computer Communication Review*, Volume 36, Number 1, pp. 17-26, January 2006.
- [22] M.L. Fisher, "The Lagrangean Relaxation Method for Solving Integer Programming Problems," *Management Science*, Volume 27, Number 1, pp. 1-18, January 1981.
- [23] M.L. Fisher, "An Application Oriented Guide to Lagrangean Relaxation," *Interfaces*, Volume 15, Number 2, pp. 10-21, April 1985.
- [24] A.M. Geoffrion, "Lagrangean Relaxation and its Use in Integer Programming,"

*Mathematical Programming Study*, Volume 2, pp. 82-114, 1974.

[25] M.S. Bazaraa, H.D. Sherali, and C.M. Shetty, “Lagrangian Duality and Saddle Point Optimality Conditions”, *Nonlinear Programming: Theory and Algorithms*, 2<sup>nd</sup> Edition, pp. 199-242, John Wiley & Sons, Inc, Singapore, 1993.

[26] S. Martello & P. Toth, “Upper Bounds and Algorithms for Hard 0-1 Knapsack Problems,” *Operations Research*, Volume 45, Number 5, pp. 768-778, September 1997.



## 簡歷

姓 名：曾中蓮

出生地：台灣省台北市

生 日：中華民國七十一年三月二十三日

學 歷：八十九年九月至九十三年六月

國立中央大學資訊管理系學士

九十三年六月至九十五年七月

國立台灣大學資訊管理研究所碩士



