國立臺灣大學資訊管理研究所

碩士論文

Graduate Institute of Information Management

National Taiwan University

Master Thesis

考慮服務品質需求下達到資訊遺漏最小化之近似最佳化機密分享與防禦資源配置規劃

Near Optimal Secret Sharing and Defense Resource Allocation Plans for QoS Constrained Information Leakage Minimization

陳冠瑋

Guan-Wei, Chen

指導教授：林永松、祝國忠 博士

Advisor: Frank Yeong-Sung Lin, Kuo-Chung Chu, Ph.D.

中華民國九十八年七月

July, 2009

考慮服務品質需求下達到資訊遺漏最小化之近似最佳化機密分享與防禦資源配置規劃

Near Optimal Secret Sharing and Defense Resource Allocation Plans for QoS Constrained Information Leakage Minimization

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：陳冠瑋 撰

中華民國九十八年七月

謝誌

在研究所求學的這兩年期間，經過許多挫折與學習，讓我對人生的歷練有進一步的瞭解，堅持到底必定可以迎向成功之路。求學應有的態度與為人處事方式亦是在我求學生涯獲得最大的收穫。這篇論文的完成，要歸功與許多給我支持幫助的師長與朋友們，請容我與你們一一道謝。

首先，要將這篇論文獻給我的父母親：陳銘勳先生與邱瑞鶯女士，您們總是時時刻刻關心我的論文進度，看到我研究上有挫折時給予我最大的支持，讓我沒有經濟上的後顧之憂全力衝刺論文。也感謝我的弟弟陳冠儒，在論文最繁忙時幫我處理家事工作，多虧有你們作為我的後盾，才得以完成我的學業。

最感謝我的恩師林永松老師，在研究這方面，您總是細心的給予我指導，毫無保留地提供我您所知的專業知識，協助我完成論文的研究，並且告知我遭遇困難時，所需具備的研究態度與精神，在這過程當中培養出嚴謹小心的個性，這是我這輩子難忘的珍寶;感謝恩師祝國忠老師，您總是在商軟下課後，陪我留下來討論論文與數學模型，給我改進的建議並且教導我 LR 的精隨，您讓我在論文新手上路的這個階段走的較順遂，感謝您的鼓勵與教誨；謝謝口試委員林盈達博士、傅新彬博士、呂俊賢博士與鐘嘉德博士對於這篇論文的建議與指教，讓此篇論文可以更加完善。

感謝博士班的學長姐，總是在我徬徨時候，提醒我論文進度的發展，讓我不敢隨意鬆懈，適時的從黑暗中重新振作站起來；接著我想對實驗室的好伙伴:瑞羅、歪歪、福福、阿保、李德與猷猷，給我們 OP 七俠自己個掌聲，真的撐過這個大難關，好幾次的挫折與失敗打擊著我們，但卻不能使我們動搖，多虧有你們的鼓勵與支持，度過這刻骨銘心的兩年生活，將成為我永生難忘的回憶，非常感謝你們大家；感謝子雋總是相當熱心的關心我，甚至口試那天友情客串，來幫我們助陣加油；感謝學弟怡緯、永斌、世昌與耀元，你們在口試那天的幫忙，使我可以專心的準備論文口試報告；感謝那些國中同學不斷地鼓勵支持，有你們會才能更堅定的步向這個終點；最後我想要把感謝的重頭戲留給女朋友小羽，妳就像我的家人一樣重要，總是在旁邊默默的支持我鼓勵我，為我打理好論文以外瑣事，妳的用心促使我的決心，只想對妳說：有妳在真好。

感謝老天給我這個機會可以進入台大資管所來學習，保佑我身體健康、保佑我研究順利，保佑我的一切。這邊獲得的所有事物的經驗，對我的未來都有著很重要的影響。我將抱持著這份態度與精神，面對未來所會遭遇到的挑戰。

陳冠瑋 謹識
于台大資訊管理研究所
民國九十八年七月

論文摘要

論文題目：考慮服務品質需求下達到資訊遺漏最小化之近似最佳化機密分享與防禦資源配置規劃

作者: 陳冠瑋　　　　　　　　　　　　　　　　　　　　　　九十八年七月

指導教授: 林永松、祝國忠 博士

資訊系統與網際網路的興盛，促使多數企業應用資訊技術來獲得競爭優勢，而各企業皆有機密的營運資料，利用電子數位化的方式儲存。但是核心營運的方針若被對手得知，將使企業失去競爭力與形象受損。因此，個人或是企業需要降低機密資訊遺漏的風險，也需考量確保機密資訊的可用性，是否能讓合法使用者在有效時間內使用機密資訊。面對日益攀升的資料竊取行為所帶來嚴重損失，發展有效的防禦策略是當務之急的議題。

本論文中提出整合網路規劃的資訊安全管理問題，將攻防情境轉化成最小-最大化的雙層數學模型問題。在內層問題中，攻擊者必須利用有限的資源來進行資料竊取動作並造成最大化傷害，包含竊取機密資訊拼圖與相對應解密鑰匙才能構成資訊遺漏的傷害。而在外層問題中，網路管理者妥善分配其預算資源，以網路規劃觀點建置拓墣，在縱深防禦概念下設計出高度強韌的網路，配合秘密分享機制與防禦資源部署達到資訊隱密性與可用度，使攻擊行為不同傳統方式，如攻克節點就可造成傷害，試圖最小化資訊遺漏的傷害損失。除此之外，因考量真實網路環境會發生的傳輸連結故障，需在網路規劃時確保整體網路傳輸的可靠度以及滿足使用者服務品質要求。針對此雙層數學問題，我們提出拉格蘭日鬆弛法及次梯度法為基礎的演算法來解決問題。另外，我們針對初始部署問題建置一個獨立單層數學模型，定義機密資訊離散指標來衡量攻擊者的影響，利用模擬退火法基礎的演算法進行處理，並利用電腦實驗來評估這些演算法的效率與效果。

關鍵詞: 資訊安全、網路規劃、秘密分享、服務品質、最佳化、資源配置、可靠度、存活度、拉格蘭日鬆弛法

THESIS ABSTRACT

GRADUATE INSTITUTE OF INFORMATION MANAGEMENT NATIONAL TAIWAN UNIVERSITY

NAME: GUAN-WEI CHEN MONTH/YEAR: JULY 2009

ADVISER: YEONG-SUNG LIN, KUO-CHUNG CHU

Near Optimal Secret Sharing and Defense Resource Allocation Plans for QoS Constrained Information Leakage Minimization

With the rapid prosperity of information systems and the Internet, most enterprises obtain competitive advantage by means of these information technologies. Hence, each enterprise uses the electronic equipment to store the sensitive information about core competence of the business. However, if the business secrets are leaked by opponents, it would lead to lose the competence and ruin their reputation for victims. For this reason, individuals or enterprises must protect the secrets from information leakage and ensure the availability for each legitimate user. As a result of the more criminal problems as time goes by, it becomes one of the important issues to develop effective defense strategies against information theft nowadays.

In this thesis, we consider the network planning in the realm of the information security. The attack-defense scenario is formulated as the min-max mathematical model. In the inner problem, the attacker must allocate his/her limited attack budget to steal the sensitive information in order to cause maximal damage. In addition, the attacker could not reveal the secret unless he/she collects the enough number of shares and the corresponding decrypted key.

On the other hand, in the outer problem, the network operator must construct the network topology and take account of the concept of defense-in-depth to design the most robust network. Furthermore, the combination of the secret sharing scheme and defense resource allocation strategy is applied for the sake of the confidentiality and availability. However, the attacker’s behavior is different from traditional attacks that he/she causes damage as soon as compromising nodes. Because of the consideration of the link malfunction, the network operator should not only guarantee the reliability of the network transmission but also satisfy the Quality-of-Service for legitimate users.

The Lagrangean Relaxation-based algorithm and the subgradient-based algorithm are proposed to solve the two layer mathematical problem. Beside, we further formulate the independent single layer model for the initial network deployment problem and define the “Discrete Degree” metric to represent the impact of the attacker. The Simulated Annealing-based algorithm is applied to handle this problem. Finally, we evaluate the efficiency and effectiveness of the proposed algorithms by computational experiments.

Key Words: Information Security, Network Planning, Secret Sharing, Quality of Service, Optimization, Resource Allocation, Reliability, Survivability, Lagrangean Relaxation Method

Table of Contents

論文摘要 IV

THESIS ABSTRACT .... **V**

Table of Contents .... **VII**

List of Tables .... **IX**

List of Figures .... **X**

Chapter 1 Introduction **1**

1.1 Background.. 1

1.2 Motivation 6

1.3 Literature Survey 8

1.3.1 Secret Sharing Scheme 8

1.3.2 QoS Routing 12

1.3.3 Survivability 16

1.4 Proposed Approach 18

1.5 Thesis Organization.. 19

Chapter 2 Problem Formulation.. **20**

2.1 Problem Description.. 20

2.2 Problem Formulation of the NPDS Model.. 23

2.3 Problem Formulation of the ATSS Model 37

Chapter 3 Solution Approach.. **40**

3.1 Solution Approach for the ATSS Model 40

3.1.1 Lagrangean Relaxation Method.. 40

3.1.2 Lagrangean Relaxation 44

3.1.3 The Dual Problem and the Subgradient Method 49

3.1.4 Getting Primal Feasible Solutions 50

3.1.5 Summary of the Solution Approach for the ATSS Model 54

3.2 Solution Approach for the NPDS Model.. 56

3.3 The Independent Model – the DDS Model 61

3.3.1 Independent Problem Description and Formulation.................................... 61
3.3.2 The Solution Approach for the DDS Model... 63

Chapter 4 Computational Experiments 67
4.1 Computational Experiments with the ATSS Model... 67
4.1.1 Simple Algorithm 1 67
4.1.2 Simple Algorithm 2 69
4.1.3 Experiment Environment.. 71
4.1.4 Experiment Results... 73
4.1.5 Discussion of Results 81
4.2 Computational Experiments with the NPDS Model.. 84
4.2.1 Experiment Environment.. 84
4.2.2 Experiment Results... 86
4.2.3 Discussion of Results 88
4.3 Computational Experiments with the DDS Model.. 89
4.3.1 Experiment Environment.. 90
4.3.2 Experiment Results... 91
4.3.3 Discussion of Results 96

Chapter 5 Conclusion and Future Work.. 97
5.1 Conclusion.. 97
5.2 Future Work.. 99

References 102

List of Tables

Table 1-2 Composition QoS Routing Problem 15

Table 1-3 The Key Properties of Survivable Systems.. 17

Table 2-1 Assumptions and Description of the NPDS Model.. 28

Table 2-2 Given Parameters of the NPDS Model... 30

Table 2-3 Decision Variables of the NPDS Model 31

Table 2-4 Given Parameters of the ATSS Model... 37

Table 2-5 Decision Variables of the ATSS Model 38

Table 3-2 Heuristic_LR_Algorithm.. 51

Table 3-3 Find_ Target_Secret Algorithm 54

Table 3-4 Lagrangean Relaxation Algorithm.. 54

Table 3-5 The Heuristic_NPDS Algorithm... 58

Table 3-6 The Secret_Adjustment Algorithm 59

Table 3-7 The Share_Reallocation Algorithm.. 59

Table 3-8 Topology_ Adjustment 60

Table 3-9 Defense_Resource_Adjustment.. 61

Table 3-10 Additional Decision Variable in (IP 3) 62

Table 3-11 SA-based Heuristic Algorithm for the DDS model 66

Table 4-1 SA_1 Algorithm 68

Table 4-2 SA_2 Algorithm... 69

Table 4-3 Experiment Parameter Settings for the ATSS Model 72

Table 4-4 Experiment Results of Small-sized Networks ($|N_I|$= 25)............................. 73

Table 4-5 Experiment Results of Medium-sized Networks ($|N_I|$= 64).......................... 75

Table 4-6 Experiment Results of Large-sized Networks ($|N_I|$ = 100).......................... 76

Table 4-7 Experiment Results under Different Defense Capability Function (B3)........ 78

Table 4-8 Experiment Parameter Setting for the NPDS Model...................................... 85

Table 4-9 The Experiment Results for the NPDS Model 86

Table 4-10 Experiment Parameter Setting for the DDS Model...................................... 90

Table 4-11 Experiment Results for the DDS model 92

List of Figures

Figure 1-1 Key Types of Incidents 3

Figure 1-2 Technologies for Information Security 5

Figure 1-3 Percentage of IT Budget for Security 6

Figure 1-4 General Threshold Scheme 11

Figure 1-5 Multicast Routing Constraints 13

Figure 2-1 Select the Position of Servers 25

Figure 2-2 Construct Network Topology 25

Figure 2-3 Shares and Decrypted Keys Distribution 25

Figure 2-4 Quality of Service Routing for Each User 26

Figure 2-5 The Reliability Verification 26

Figure 2-6 Defense Resource Allocation 26

Figure 2-7 Attacking Targets 27

Figure 3-1 Concepts of the Lagrangean Relaxation Method 41

Figure 3-2 Lagrangean Relaxation Method Procedure 43

Figure 3-3 Solution Approach for the NPDS Model 56

Figure 3-4 The Procedure of Simulated Annealing Method 65

Figure 4-1 The Network Vulnerability under Different Numbers of Users ($|N_I|$= 25) .. 79

Figure 4-2 The Network Vulnerability under Different Numbers of Users ($|N_I|$= 64) .. 79

Figure 4-3 The Network Vulnerability under Different Numbers of Users ($|N_I|$= 100) 80

Figure 4-6 The Improvement under Different Reallocation Strategies ($|N_I|$= 25) 87

Figure 4-7 The Improvement under Different Reallocation Strategies ($|N_I|$ = 64) 88

Figure 4-8 The Improvement under Different Reallocation Strategies ($|N_I|$ = 100) 88

Figure 4-9 The procedure of the second solution approach 90

Figure 4-10 The Improvement of Top 10 Solutions under Different Network Sizes (Uniform Distribution) 93

Figure 4-11 The Improvement of Top 10 Solutions under Different Network Sizes (Normal Distribution) 94

Figure 4-12 The Improvement of Top 10 Solutions under Different Network Sizes (Deterministic) 95

Chapter 1 Introduction

1.1 Background

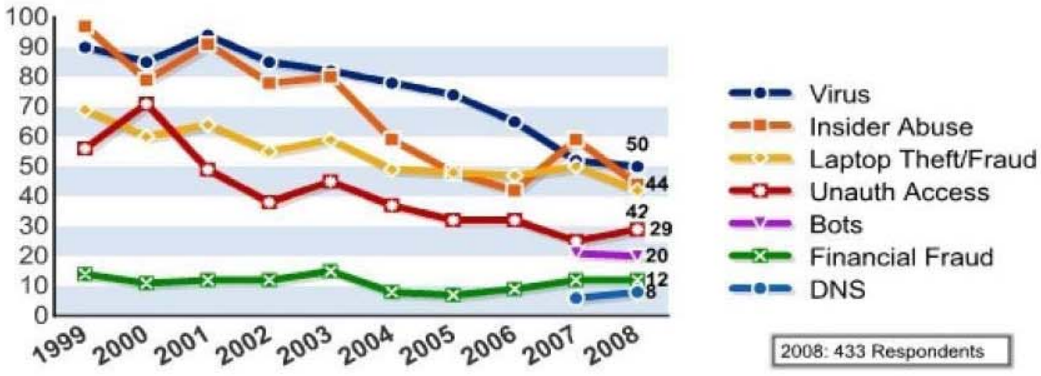
Computers of nowadays become vital roles for us to work, to shop, to recreate and so on so forth. According to Moore's Law, which is proposed by Intel co-founder Gordon Moore [1] in 1965, describes that the number of transistors on a chip will double about every eighteen months. The meaning of that is information technology has been more capable of dealing with people's daily lives, and the storage devices also cost down every few months [1]. Moreover, the rapid growth of Internet makes many individuals, schools and enterprises generate a great deal of demands. Therefore, most enterprises transform their daily work into using IT to process, which the prospect not only increases the efficiency of business practices many times than before but also assist them deal with real time problems. For example, www.bandongo.com is the famous sharing space for free or premium members through the Internet.

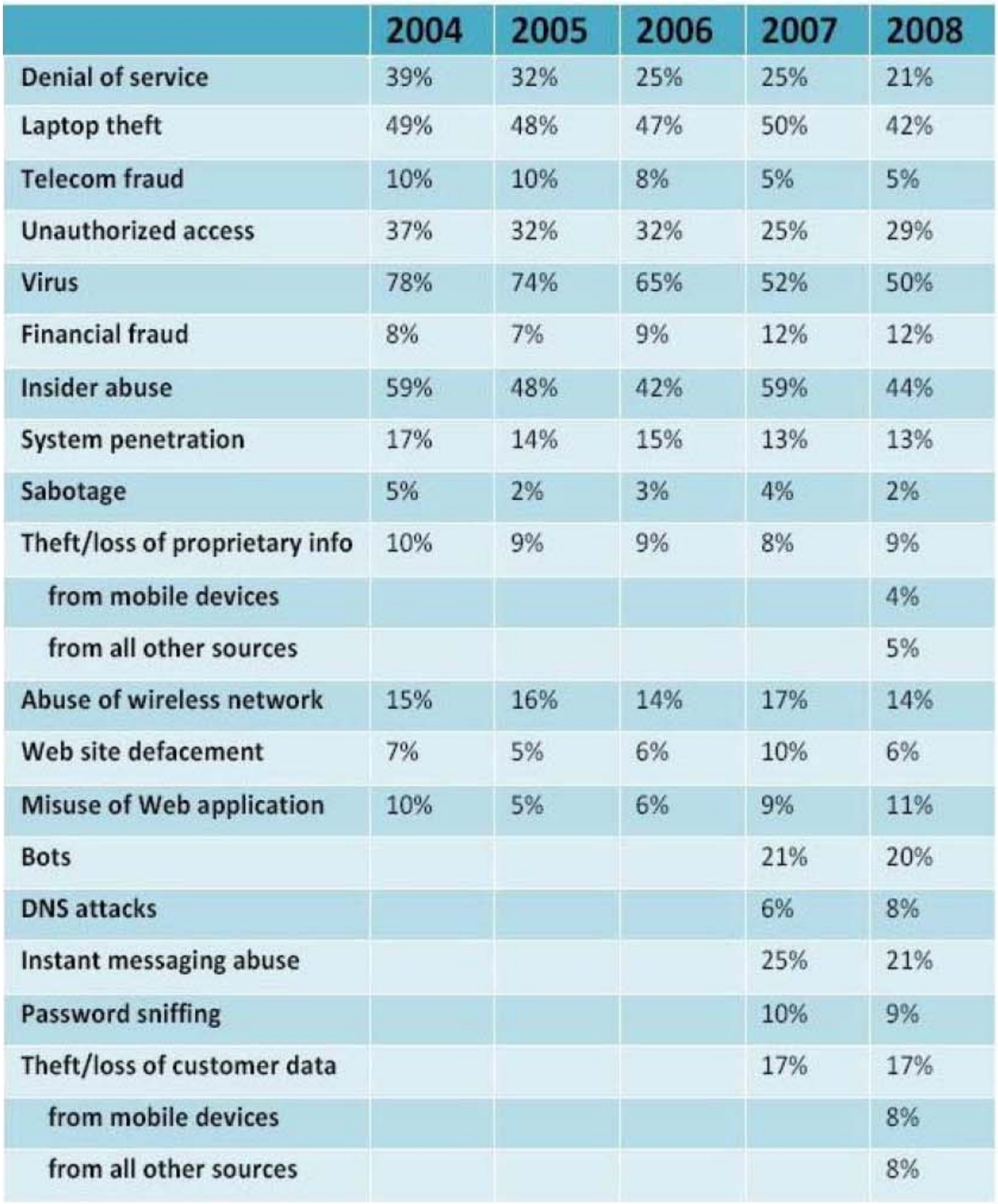
On the whole, the enterprises must provide information sharing mechanisms for internal users to access securely. However, the upgrade of technology still brings the bad side effect, which is computer crime events increasing rapidly [2]. It is as the saying goes: "water can either float or turn over a boat." Hackers apply a variety of tools to steal information for fun or gaining benefits, called cyber-crime, whose range are from injecting worms, Trojan horse, backdoor program to web phishing. Consequently, the cyber-crime events have become urgent problems for network security to solve during these years recently.

Information theft, Distributive Denial of Service (DDoS) and viruses are top cyber-crime issues in the recent years, shown in Figure 1-1. What is more, we could observe more types of criminal behaviors or incidents in 2008 than other years. According to the CSI/FBI Computer Crime and Security Survey (2008) [3], the information leakage already caused serious damage and loss for most enterprises. The damage of information leakage is unlike others cyber-crime attacks that hackers intrude our computers or networks to steal information quietly so as to avoiding being found by the network operator. It seems not to happen to any unusual phenomenon until they announce or publicize the stolen information. Consequently, we attach the great importance to information security issues more than before in order to lower the effect of criminal behaviors and incidents.

Typically, the information system should provide the continuous service for all legitimate users to satisfy reasonable Quality of Service (QoS) requirements even though it suffers from intentional attacks or natural accidents. Malicious attacks might cause serious tangible and intangible damage for victims as financial loss, ruined his/her reputation respectively [2]. There are a number of security tools against intentional attacks and the several authentication and authorization mechanisms, shown in Figure 1-2. It is easy to observe from Figure 1-2 that almost more than eighty percentage the enterprises install anti-virus software, firewalls, and virtual private network in addition to raise gradually the percentage from 2006 to 2008.

Although no one could protect systems from attacking perfectly, the network operator could adopt some strategies to reduce the probability of cyber-crime events. In other words, a so-called one hundred percent of the security system never exists in the real world [2]. The author points out [3] few budgets applied on the IT department, which 53 percent organizations allocated less 5 percent of over IT budget, shown in Figure 1-3. Nevertheless, enterprises must invest more and more resources including money, labor, power, time, and network deployment to strengthen the robustness of the system if they anticipate reducing the risk of cyber-crime events effectively.



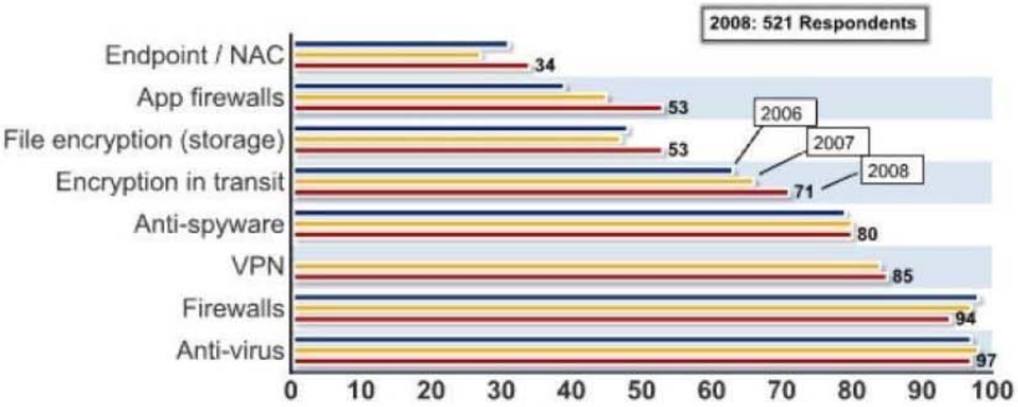


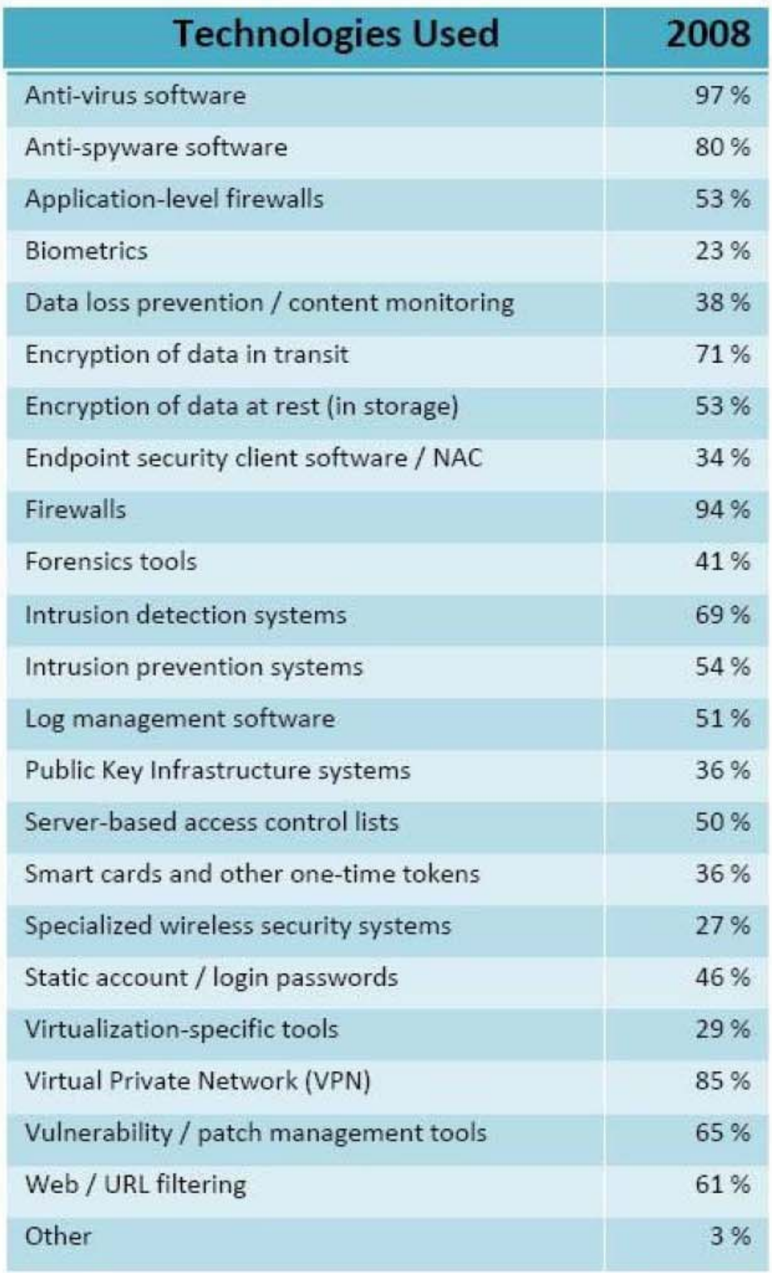
	2004	2005	2006	2007	2008
Denial of service	39%	32%	25%	25%	21%
Laptop theft	49%	48%	47%	50%	42%
Telecom fraud	10%	10%	8%	5%	5%
Unauthorized access	37%	32%	32%	25%	29%
Virus	78%	74%	65%	52%	50%
Financial fraud	8%	7%	9%	12%	12%
Insider abuse	59%	48%	42%	59%	44%
System penetration	17%	14%	15%	13%	13%
Sabotage	5%	2%	3%	4%	2%
Theft/loss of proprietary info	10%	9%	9%	8%	9%
from mobile devices					4%
from all other sources					5%
Abuse of wireless network	15%	16%	14%	17%	14%
Web site defacement	7%	5%	6%	10%	6%
Misuse of Web application	10%	5%	6%	9%	11%
Bots				21%	20%
DNS attacks				6%	8%
Instant messaging abuse				25%	21%
Password sniffing				10%	9%
Theft/loss of customer data				17%	17%
from mobile devices					8%
from all other sources					8%

Figure 1-1 Key Types of Incidents

In recent years, people access necessary data and information electronically through digital storage so frequent that the confidentiality, reliability, availability, integrity of the data storage device becomes the importance of information security [4]. For instance, RAID and SAN are chosen for information sharing, if network operators further consider privacy issues, they will encrypt data by private or public key to transmit. However, there are some problems for the available considerations here. If random accident failures occur on the critical nodes, the information on those nodes will lose resulting in data inaccessible. Beside, in other cases, if intruders might get the encrypted key, they could leakage confidential data which one they desire as well.

To minimize the damage of information leakage, we should apply many the combinative methods of security strategies and defensive plans simultaneously to deal with network security problems rather than single mechanism. For the former reason, it is the better solution for the essence of the security issues to construct the robust network topology. In our thesis, the character of our targeted system provides users can securely store critical information to ensure the persistence, to be continuously accessible, to not be destroyed, and to keep confidential [5]. It's so called "Survivable Storage Systems", which must guarantee over time in spite of occurring malicious attacks and random errors side by side.





Technologies Used	2008
Anti-virus software	97 %
Anti-spyware software	80 %
Application-level firewalls	53 %
Biometrics	23 %
Data loss prevention / content monitoring	38 %
Encryption of data in transit	71 %
Encryption of data at rest (in storage)	53 %
Endpoint security client software / NAC	34 %
Firewalls	94 %
Forensics tools	41 %
Intrusion detection systems	69 %
Intrusion prevention systems	54 %
Log management software	51 %
Public Key Infrastructure systems	36 %
Server-based access control lists	50 %
Smart cards and other one-time tokens	36 %
Specialized wireless security systems	27 %
Static account / login passwords	46 %
Virtualization-specific tools	29 %
Virtual Private Network (VPN)	85 %
Vulnerability / patch management tools	65 %
Web / URL filtering	61 %
Other	3 %

Figure 1-2 Technologies for Information Security

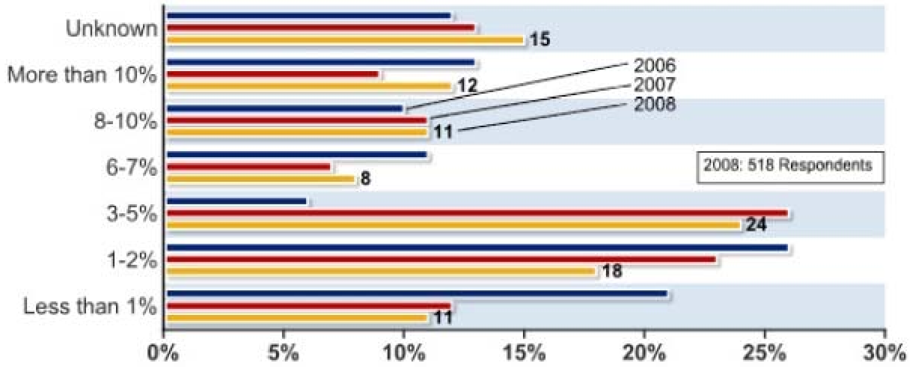


Figure 1-3 Percentage of IT Budget for Security

The combinative method of secret sharing and replication mechanisms achieves our goals, which are the confidentiality and availability requirements. We consider both the depth of deployment and the width of deployment to handle tradeoff, further discussing in section 1.3. In addition, we adopt the concept of the network planning to design the network topology and to satisfy Quality of Service (QoS) requirements for users. To assume the attacker is extremely excellent, we consider the worst case in this mathematical problem. That is the attacker always can find the most efficient ways to maximize system damage, but the network operator also can apply the appropriate strategies to reduce system damage and to maintain the specific level of system performance. Both two parties would adjust their strategies dynamically to reach the optimal strategy until the network operator obtains the strategy which achieves the minimal network vulnerability.

1.2 Motivation

The ubiquitous Internet makes the growth of cyber-crimes, and one of the most serious cyber-crimes is information theft, which the attack behavior is easily ignored because it does not alert the victim but causes inestimable damage instead; moreover, information leakage incidents have accumulated rapidly during these years recently. Accordingly, it causes and brings the serious threat to individual privacy, property loss or financial loss, even jeopardizing the security of our nation [6]. To take into account that not only external attackers intend to exploit the vulnerabilities of the system and steal its information, but also there are internal natural disasters and intentional destructions to make its information unavailability leading to authenticated users not accessing function normally. Typically, the enterprise with replication mechanism could spend less time to recover business processing and restart to provide the service if it encounters the critical catastrophe. Take the 911 terrorist attacks as example, some suffered enterprises could minimize the impact of the system crashed and restructure within the short period because they consider the opportunity cost to adopt replication or backup mechanism.

According to above these issues, it is requisite for us to share information under real distributed systems in fact. The network operator must consider both the system confidentiality and availability aspects into system performance; otherwise it would be attractive for intruders to attack. There are many cases about information leakage events, which we name them “system damage” in our research. Many information leakage events in [6], as American nation claims that Russian hackers intrude their military network to steal the sensitive information of the national defense; the bank of America-National Trust & Association lost its magnetic tapes, which store one hundred twenty million records of governmental employees, resulting in extreme damage and ruin inestimable reputation to the system administrator and the network operator.

Even though a lot of drawbacks and risks exist in the network, most enterprises still store and share data and information through the network system. Some problems are derived from information backups, recoveries, and sharing the information between legitimate users, while we tackle these problems then try to keep the information more secure. It is quite important to incorporate the optimal protection parameters into the defense optimization problem [7]. To achieve the information security, we apply the technology of secret sharing for confidentiality [5][8][9][10], then processing the method of replication [11] for availability, which both of them are the tradeoff. Besides, there is the failure probability of links occurring in our targeted system for the sake of corresponding with the real environment. As a result, we must also consider reliability requirements as an impact factor to construct the network topology. The same concept is proposed by Levitin in 2007: "Optimization of system structure for systems developed from scratch [7]." The network operator must guarantee the legitimate users of the QoS routing mechanism [12][13][14] so that they could receive enough shares to recover within the reasonable time.

The attacker allocates limited budget appropriately to construct the attack tree which can maximize system damage. On the other hand, the network operator adopts the efficient strategy and invests limited resource to enhance the robustness of the whole network in order to minimize the damage of attackers. Our model combines the optimization of system performance and defense measures to reduce the expected system damage as well as being considered this concept in [7].

According to realm of attack-defense scenario, we implement to construct the network topology because the topological structure is the important factor affecting defense-in-depth of the network. We consider this state as the mathematical problem to describe between network operators and attackers. Accordingly, we propose two-level mathematical optimization problem and solve it with our proposed solution approaches.

1.3 Literature Survey

1.3.1 Secret Sharing Scheme

Secret sharing schemes are the cryptography techniques where the sensitive information is encoded into several fragments by public or private key mechanisms, called shares, such that certain combinations of shares can together recover the encoded secret. This concept was firstly proposed by Adi Shamir and George Blakley in 1979 respectively [8]. This schemes are also called *(t, n)*-threshold schemes. It is meaning that intruders get *t-1* shares given no information on encoded secret until they receive enough *t* shares. If we divided the single secret into the more shares, it will need more storage capacity. However, the constraint of *(t, n)*-threshold scheme is $({}^{n}\!/_{2}+1) \le t \le n$. The secret is more confidential if the threshold *t* is set higher, but it would cause insufficient to use. Besides, Martin Tompa and Heather Woll [9] further proposed the verification mechanism so as to handle the existence of cheaters under traditional secret sharing scheme. Dealers split the secret into *N* shadows and distribute them through the secure channels to each participant.

The other famous information dispersal scheme is generalized secret sharing scheme proposed by Ito, Saito, and Nishizeki that implemented any access structure with the *(q, q)* -threshold secret sharing scheme. In [11], the author shows the scheme how to work below. Consider a set of *r* participants $\{P_1, P_2, ..., P_r\}$ such that any *m+1* participants can reveal the encoded secret. Denote the set $B = \{B_1, B_2, ..., B_q\}$ consists of possible combinations of *m* participants, and determine $q = C^{r}_{m}$ to set *(q, q)*-threshold. Generate shares $S = \{S_1, S_2, ..., S_q\}$, and they are assigned to participants P_i by the function $g(i) = \{S_j, P_i \notin B_j, 1 \le j \le q\}$. Each share is stored at most *r-m* participants, in the meantime, each participant only receive no more than C^{r-1}_{m} shares. For example, assume a set of four participants needs at least three participants, and then $r= 4$, $m=2$ and $B= \{(P_1, P_2), (P_1, P_3), (P_1, P_4), (P_2, P_3), (P_2, P_4), (P_3, P_4)\}$. Next, six shares are produced $\{S_1, S_2, S_3, S_4, S_5, S_6\}$, all of them needed to recover the secret. Assign six shares by function:

Participant P_1 receive shares $\{S_4, S_5, S_6\}$, Participant P_1 receive shares $\{S_2, S_3, S_6\}$, Participant P_3 receive shares $\{S_1, S_3, S_5\}$, Participant P_4 receive shares $\{S_1, S_2, S_4\}$.

Compared to *(t, n)*-threshold secret sharing mechanisms, the *(q, q)*-threshold one can provide more confidential, but it needs a large number of storage space. Above two secret sharing schemes are essential and chosen them according to the requirements of the systems. In [8][9][10] implied some considerations to design secret sharing scheme as following:

1. How to solve the problem of fault dealers and exclude from legal groups.
2. How to find fault shares during the reconstructive period.
3. How to design fair mechanisms in order to avoid some participants taking advantage to cheat others
4. How to distribute the reusable shares to recover multiple secrets.
5. How to solve dynamic secret sharing, that is the secret must not change and redistribute when new participants attend.

There are types of the secret sharing application, verifiable secret sharing which could detect dishonest dealers, proactive secret sharing which could periodically update the share without changing the original secret, quantum secret sharing which could find out the existence of eavesdroppers, and multi-secrets sharing scheme which could allow parallel reconstruction. Nowadays, the popular usage of secret sharing application is to design image protection and watermarking in order to achieve higher information security.

The Figure 1-4 shows the secret sharing of some of the general threshold scheme in the recent years [5]. We call Figure 1-4 (a) is "Replication *(1, n)*" mechanism, which increases information availability but provides no information confidentiality because of entire copy of data. Figure 1-4 (b) is “Splitting *(n, n)*” which provides perfect confidentiality but least availability because all shares needed to recover. In addition, if one share is lost or destroyed, the data could not be recovered to become useless. Figure 1-4 (c) names “Decimation *(1, n, n)*”, which feature divided information into *n* shares, if the intruders get one share, he would expose *1/n* of the secret, so it offers no information-theoretic confidentiality. Figure 1-4 (d) is “Rabin’s information dispersal algorithm *(1, m, n)*” mechanism, which provides the tradeoff between availability and confidentiality, but like Decimation it has no information-theoretic confidentiality too.

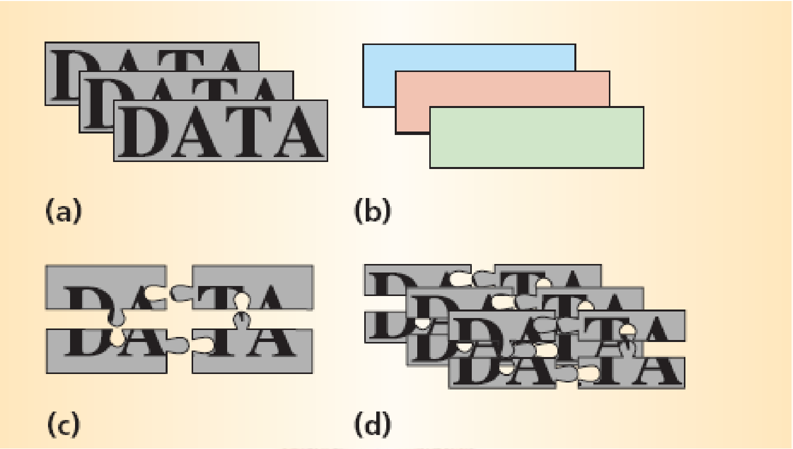


Figure 1-4 General Threshold Scheme

In our research, we consider for fault tolerant and more secure data storage service. Therefore, we use *(k, n)* threshold secret sharing for data confidentiality and availability, and adopt the replication-based mechanism to avoid the data inaccessible for users. Moreover, we combined threshold schemes and cryptographic techniques to encrypt the original information with random keys, and store encrypted information with secret sharing. This technique derived from the concept of “short secret sharing” further to enhance the confidentiality of information. Finally, we must confirm that all authenticated users can receive the sensitive information during the maximal tolerate time, the same concept shown in [11].

1.3.2 QoS Routing

Today, we consider the most demanding application as an interactive or time limit application from the service quality point of view. However, based on Internet, it only supports the datagram service which is called “best effort.” That is to say, the routing mechanism of the Internet tries to its best to forward traffic, and it can’t guarantee regarding end to end delay, delay jitter, packet loss rate, bandwidth, etc [15]. For this reason, QoS routing extends from current routing [12][13][14] that it transports data using integrate-service class of service, calculating routing metrics such as delay and residual bandwidth between node pairs of multiple paths. In addition, today’s optimal path routing algorithm can’t change alternate paths that the new flow is not admitted even if there is an adequate alternate path. In contrast, QoS routing can shift the traffic to the better path as soon as such the path exists.

Multicast routing mechanism usually combines several QoS constraints to achieve possibly requirements, such as end to end delay, minimum bandwidth, delay jitter, or a combination thereof for groups. Multicast routing based on current network states and topology to construct multicast tree in order to optimize the objective function [12]. The components consisted of the multicast routing showing Figure 1-5. The multicast tree represents the reachable path from the source to the destination and on this route satisfying specific QoS merits constraints.

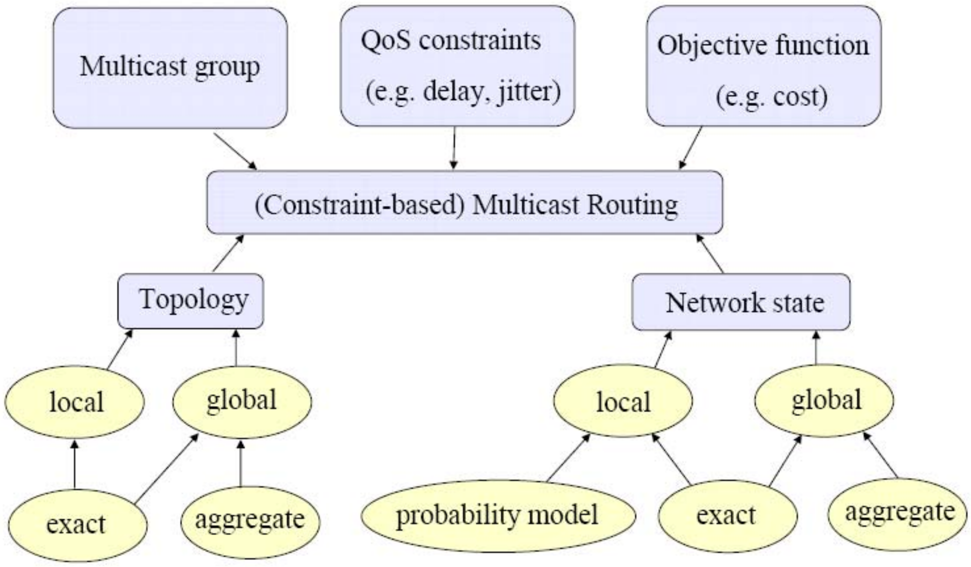


Figure 1-5 Multicast Routing Constraints

Chen and Nahrstrdt [14] express QoS requirements as a set of constrains such as link constrains, path constrains, and tree constrains. They use the basic routing functions to solve composite problems and to find feasible paths or trees, presented in Table 1-2. Link constraints are the route selection under the restriction of the link selection, such as buffer or bandwidth on the link; tree constraints are under the multicast tree of the restrictions, such as providing different delays for each type of users. These authors proved that any combination QoS routing of the tree optimization or the multiple multiplicative constraints is the NP-complete problem. There are three basic composition rules of tree constraints as following [13].

Denoted *m (i, j)* as metric for link *(i, j)*, any path *p* as *(i, j, k, …, x, z)*.

- Additive tree constraints: The metric is cost, delay, and delay jitter following this composition rule. Delay of all links sum up equal to end to end delay from the source to the destination. The end to end delay from node *i* to node *z* could be formulated the mathematical form as

$$d\ (p) = d\ (i,\ j) + d(j, k) + \ldots d\ (x, z).$$

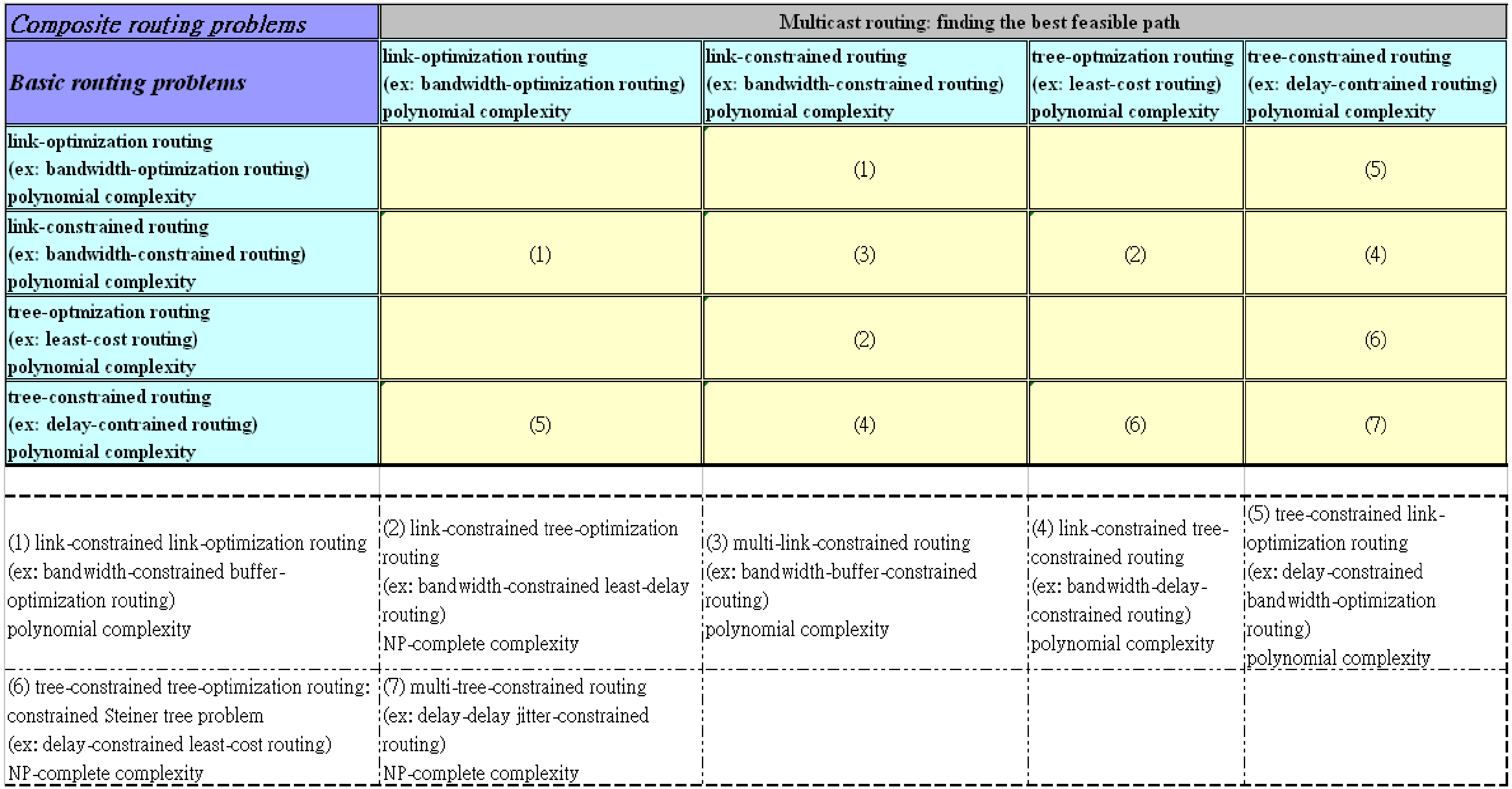
- Multiplicative tree constraints: The metric is the probability of successful transmission. The OD pair path can transmit data if the all nodes on the path are not failure. The probability of successful transmission could be formulated the mathematical form as

$$t(p) = t(i, j) \times t(j, k) \times \ldots t(x, z).$$

- Concave tree constraints: The bandwidth of the metric follows this composition rule. The bottleneck of the path is determined by the minimal channel. The bandwidth of the path could be formulated the mathematical form as

$$bw(p) = min\{bw(i, j), bw(j, k), \ldots, bw(x, z)\}.$$

We use these rules above to satisfy the specific QoS requirement in our research.



Composite routing problems	Multicast routing: finding the best feasible path			
Basic routing problems	link-optimization routing (ex: bandwidth-optimization routing) polynomial complexity	link-constrained routing (ex: bandwidth-constrained routing) polynomial complexity	tree-optmization routing (ex: least-cost routing) polynomial complexity	tree-constrained routing (ex: delay-contrained routing) polynomial complexity
link-optimization routing (ex: bandwidth-optimization routing) polynomial complexity		(1)		(5)
link-constrained routing (ex: bandwidth-constrained routing) polynomial complexity	(1)	(3)	(2)	(4)
tree-optmization routing (ex: least-cost routing) polynomial complexity		(2)		(6)
tree-constrained routing (ex: delay-contrained routing) polynomial complexity	(5)	(4)	(6)	(7)

(1) link-constrained link-optimization routing
(ex: bandwidth-constrained buffer-optimization routing)
polynomial complexity

(2) link-constrained tree-optimization routing
(ex: bandwidth-constrained least-delay routing)
NP-complete complexity

(3) multi-link-constrained routing
(ex: bandwidth-buffer-constrained routing)
polynomial complexity

(4) link-constrained tree-constrained routing
(ex: bandwidth-delay-constrained routing)
polynomial complexity

(5) tree-constrained link-optimization routing
(ex: delay-constrained bandwidth-optimization routing)
polynomial complexity

(6) tree-constrained tree-optimization routing: constrained Steiner tree problem
(ex: delay-constrained least-cost routing)
NP-complete complexity

(7) multi-tree-constrained routing
(ex: delay-delay jitter-constrained routing)
NP-complete complexity

Table 1-2 Composition QoS Routing Problem

1.3.3 Survivability

Due to the prosperity of Internet, computer network becomes unbounded which characterized by managing with distributed administrative control without central authority [5]. In unbounded networks, the network operator must realize that each node in the network might be compromised that the system situates the unsafe environment. If the system must still maintain the essential services no matter which particular nodes involved, we will call such system being survivable requirements even communication between nodes are not warranted.

Organizations are dependent on the Internet network causing that the business risks are amplified because of the increment of the intrusive probability. Fortunately, we can gain the appreciation for the importance of survivability as symbiotic partner to security. It is a discipline that blends computer security with business risk management for the purpose of protecting highly distributed information services and assets [16]. Many types of survivability has been defined, and in [2][16][17] we can know the consistency that survivability is "the capability of a system to provide essential services even after successful intrusion and compromise, and to recover full services in a timely manner."

Thus, a survivable system is derived from the concept of the survivability and it must be capable to deliver essential services such as a storage data service in face of attack, failure, or accident events [18]. There are many important definitions [17] for the survivable system in terms of the tradeoff among multiple quality metrics such as performance, confidentiality, reliability, availability, fault-tolerance, modifiability, and affordability. For example, in context of survivable storage in PASIS [5], which is the tradeoff space of confidentiality, availability and performance, however, it increase confidentiality requirements to lower availability and performance.

There are three key issues to design for survivability: 1) impacts of adjustments to fault tolerate; 2) impact of security feature; 3) to determine feasible the infrastructure for given security and availability requirements [18]. Therefore, the survivable system requirements are various determined by system scope, criticality, failure, and denial of service; likewise the category of definition of the survivable system is function, use, development, and operation respectively. Table 1-3 shows the major properties of survivable systems [17].

Key Property	Description	Example
Resistance to attacks	strategies for repelling attacks	user authentication
Recognition of attacks	strategies for detecting attacks, understanding the current state, evaluating the extent of damage	recognition of intrusion usage patterns and checking integrity
Recovery essential services after attack	strategies for restoring, limiting, and maintaining compromised information within the time constraints of the mission	replication and redundancy of data or service
Adaptation to reduce effectiveness of future attacks	strategies for improving system survivability by acquiring experience from intrusions	incorporation of new patterns for intrusion recognition

Table 1-3 The Key Properties of Survivable Systems

We figure out the common feature of the survivable system that no single point failure within the whole network. Therefore, it is an important issue to apply the methodology to achieve the network continuity and minimize the impact of node malfunction or transmission failure. In [17], the author suggests some solutions for above four aspects described as the following:

- Resistance: Using traditional security, including encryption and covert channels, diversity and maximized differences in individual nodes.

- Recognition: Using intrusion monitoring, suspicious activities, system behavior and integrity monitoring.
- Recovery: Using physical and information redundancy.
- Adaptation and Evolution: Using general or specific changes to resist, recognize, or recover from new vulnerabilities that are discovered, and broadcast of adaptation and evolution strategies

We consider the damage of the revealed information as the performance metric to measure the system vulnerability in our model. In other words, the less system damage caused by attackers in the survivable system, the less the system vulnerability is.

1.4 Proposed Approach

In this research, we proposed a min-max mathematical model to formulate the outer problem which is the network planning and defense strategy problem (NPDS) and the inner problem which is attack target selecting strategy problem (ATSS). Of cause, we can not only realize that the maximal damage occurred under certain pattern of share distribution and network topology, but also find the best defense strategy for the network operator. In order to solve this two level problem optimally, Lagrangean Relaxation method and the subgradient method usually are applied to solve this highly complex problem. First, we will solve the inner problem (ATSS), and then use the result of the ATSS model as input into the outer problem. Next, we adjust the decision variable of the outer problem, and the result becomes the feedback of the inner problem. Finally, we process these above steps to find the solution iteratively. In addition, we design optimization-based heuristic algorithms to make the gap of the bound tight in order to find the fit solution. Here, we further propose the second solution approach for the initial deployment problem, called the Discrete Degree of Secret model (DDS). The constraints of the independent model are the same as the NPDS model, and the new metric in objective function represent the impact of the attacker. Then we use the Simulated Annealing method to enhance the discrete degree of secrets in order to reduce the probability of recovered information. To evaluate the result of the strategy, the vulnerability of the network is compared under the different attack-defense scenarios.

1.5 Thesis Organization

The remainder of thesis is organized as the following. In chapter 2, we describe the attack-defense scenario problem and formulate the model of the NPDS and the ATSS. In chapter 3, solution approaches to the NPDS and the ATSS problems are presented. We proposed the solution approaches based on Lagrangean Relaxation in section 3.1, and adopted the solution approach to the NPDS problem based on several mechanisms in section 3.2, including the secret adjustment, the topology adjustment and the defense reallocation adjustment. In chapter 4, the computational outcomes of the NPDS problem and the ATSS problem are presented and we further proposed the independent DDS model and its solution approach to evaluate the efficiency of our solution. Finally, in chapter 5, we presented our conclusions and in indicate possible directions of the future research.

Chapter 2   Problem Formulation

2.1 Problem Description

There are several business secrets, which enterprises make profits efficiently. Therefore, they must store the sensitive information in order to be not revealed by others, no matter various types of businesses. As a result, we both consider the risk of the information leakage and ensure the information could be used within the maximal tolerate time for legitimate users, meaning security and availability. Furthermore, we adopt the replicate mechanism for authenticated access to maintain data integrity in case of the certain file servers shut down. If the information leakage occurs, the core competence of the business will be known by their opponents, who will take suppress strategies against the victims further. The result of the victims loss their competitive advantages and also cause negative effects on their reputation.

In order to improve the consequence radically, we advocate constructing the most robust network topology, which combines the concept of defense-in-depth and satisfy quality of service requirements of authenticated users with the viewpoint of network planning. For instance, the attacker must overcome more obstacles if the network operator could construct a linear region. Although the network is extremely robust, the availability is limited. It is a tradeoff between security and availability for operators.

Since the network planning is proven NP-complete problem, it is difficult to get the solution even if we formulate the network planning as an optimization problem. The model developer should make a few significant decision variables by determining which aspects to exist in the model and which aspects to omit. As a result, we consider the some decision variables into our model, which are related information security aspects rather than aiming of all dimensions. What is more, we integrate the secret sharing scheme into our model to achieve the trade-off between confidentiality and availability [11]. The sensitive information should be encrypted before applying secret sharing scheme so as to enhance the confidentiality further. This idea is similar to the movie “National Treasure”, even evil men rob the treasure map, but they can’t find out the location of the treasury because of misunderstanding ancient writing hints. They have to inquire the expert of this domain; otherwise, it would be more difficult for them to achieve the mission.

The budget of network operator divides into two sections: one is applied to the network planning implementation; the other one is allocated defense resource on each node. Unlike the traditional network crimes, the attackers cannot bring out damage as soon as they compromised the current node while they must pay enough effort to compromise other nodes to recover the one secret. If the attacker causes damage, he should not only get enough shares, but also find the corresponding decrypted key.

To assume that one international enterprise must establish a great deal of the information centers around the world to provide service for subsidiary companies. The network operator can determine that what kind of material to be chosen for the network reliability, furthermore, we denote that all nodes are able to transmit and store the share and the decrypted key, and some legitimate users have requests in the network.

According to the distribution of all nodes that the distance and the cost of link between an arbitrary pair of nodes are given, we would construct the network topology. The network operator must take account of the existence of the link malfunction under the realistic environment. Because of the random error of links considered, the network operator could select different material types for the link to achieve specific reliability requirements. Moreover, the cost is positively related with the reliability of the normal function. Finally, we should apply budget to allocate the defense resource on critical nodes so that the risk of the leakage can descend maximal damage caused by attackers. In addition, the targeted network we discussed is an Autonomous-system (AS) level Internet. Topology is undirected graph, and each node represents domain and each link represents the inter-domain connection.

The attacker outside the AS must enter into the AS through compromising the entry node if he intends to reveal the sensitive information in the targeted network. If the attacker allots more or equal budget than nodal capability, we say this node is compromised [19]. Furthermore, the attacker constructs the attack tree or path from his initial position to the target node where all intermediate nodes on the path or tree must be compromised. Hence, the attacker uses his budget appropriately and does his best to recover the sensitive information causing maximal damage. To evaluate the effect of the attack strategy for the attacker, we define system vulnerability as the metric. Denote that S_{υ} is the value of the sensitive information, calculating the vulnerability of the network below respectively:

$$system\text{ - }vulnerability(\%) = \left(\frac{\sum_{\bar{\upsilon} \in information\ that\ is\ revealed} S_{\bar{\upsilon}}}{\sum_{\upsilon \in all\ information\ of\ the\ network} S_{\upsilon}}\right) \times 100\%$$

The less vulnerability of the network is obtained, the better strategy we proposed for network operators.

2.2 Problem Formulation of the NPDS Model

The objective of the attacker is characterized by trying his/her best to maximize the damage; in the same fashion, the network operator should minimize the damage caused by the attacker as possible. Due to such problem, we formulate it as a two layer mathematical model. To evaluate the effectiveness of the model, we assume the worst case in our scenario. It means that the attacker has perfect information about the location of decrypted keys and shares, and how is defense resource distributed in the network topology [20]. Because we consider the network at the AS level, the attacker has to compromise all nodes to reach the target on the attack path rather than attacking the arbitrary nodes.

In the NPDS model, both the attacker and the network operator has limited budget. The network operator must construct the network topology and distribute the shares and the decrypted key on nodes shown in Figure 2-1 to Figure 2-3. In Figure 2-4, the network operator verifies that each legitimate user must receive shares within the reasonable period by QoS routing.

Specifically, the artificial flows are used to ensure the connectivity between users and secrets, and we determined the number of link disjoint paths hinging on the impact of the legitimate user, shown in Figure 2-5. To ensure the legitimate user can access the server successfully, which contains the share of the secret, we use the artificial capacity and the artificial flows with the min cost flow or k-shortest path algorithms so that we can guarantee the number of the link disjoint paths. Under such restriction, each link could be used only once for each user that the reliability of the chosen path is promised at the certain risk. That is to say, the network operator tries to construct the network topology, which sustains the connectivity of the whole network system in the cause of achieving the availability and reliability.

Figure 2-6 allocates defense resource depending on the pattern of the shares and keys. On the other hand, the attacker enters an initial node *O*, and he/she probes all neighbors of *O*, allocating the more attack budget than node capability to compromise the node. However, the attacker can’t cause the damage unless he gets the enough shares and corresponding decrypted keys (Figure 2-7) to (Figure 2-8). The detail procedure is presented below.

Attack-Defense Scenario

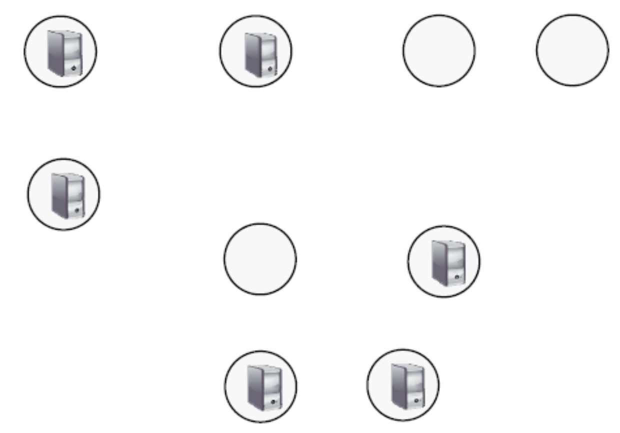


Figure 2-1 Select the Position of Servers
The network operator sets the servers which are able to store the sensitive information.

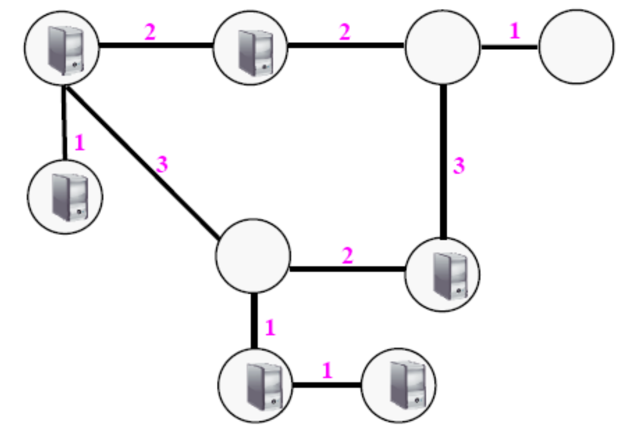


Figure 2-2 Construct Network Topology
The network operator depends on the distance between nodes to determine which link to set with different material types.

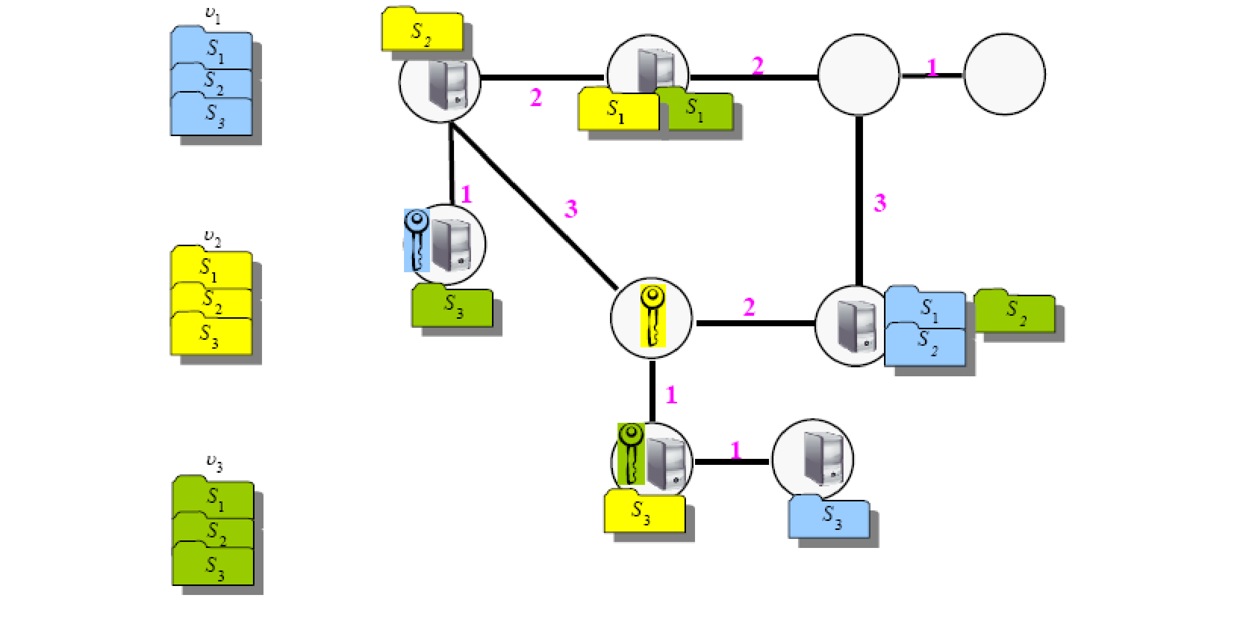


Figure 2-3 Shares and Decrypted Keys Distribution
Network operators must design different patterns to distribute the shares and corresponding decrypted keys to the appropriate position.

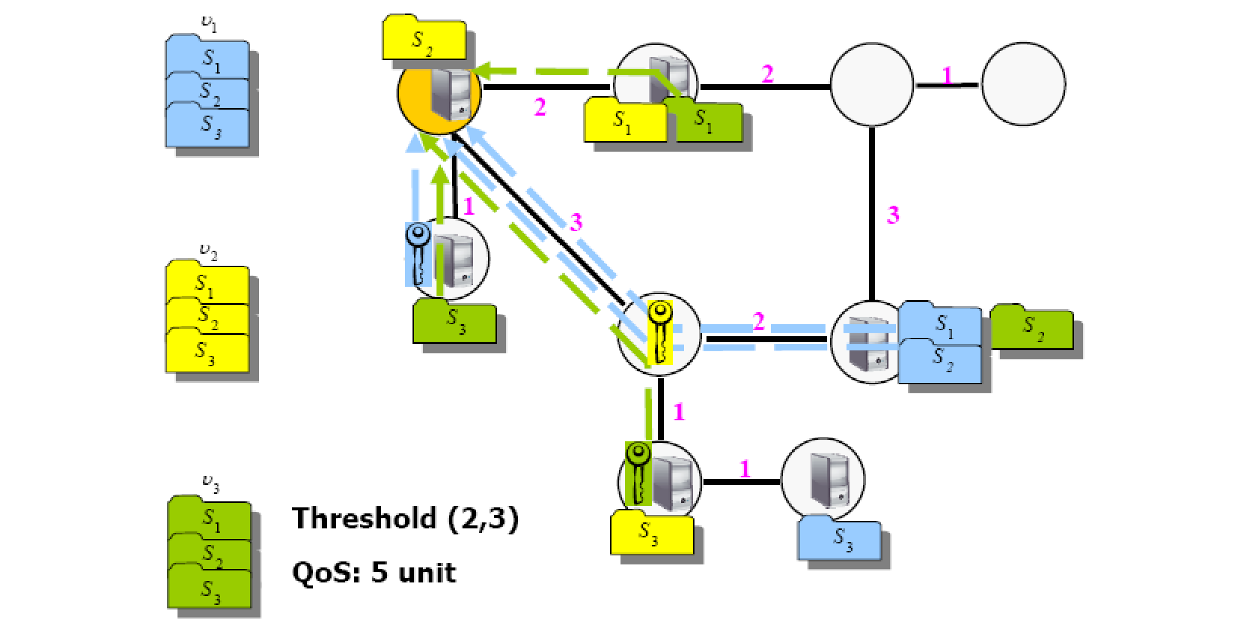
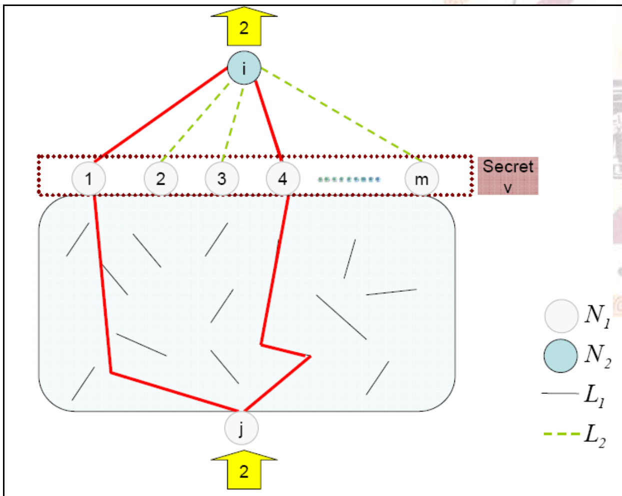


Figure 2-4 Quality of Service Routing for Each User
The user must get at least two piece to recover sensitive information during tolerate maximal time. In addition, the users also must obtain the corresponding key to decrypt. In this case, the threshold of the secret sharing is (2, 3) and tolerate maximal time for each secret is five unit.



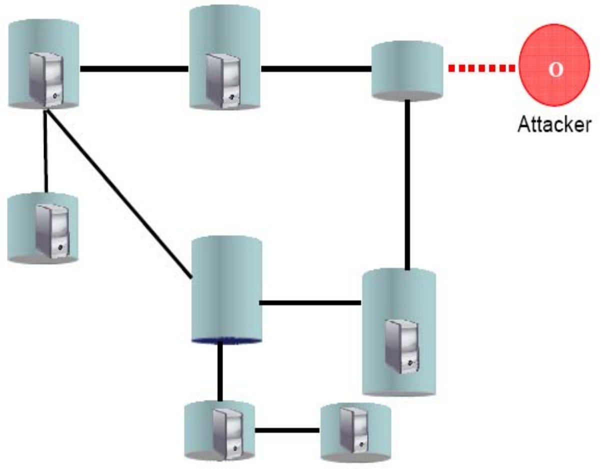


Figure 2-5 The Reliability Verification
The network operator applies the artificial flows to ensure that there are one or more link disjoint paths from the dummy node *i* to the user *j*. The dummy node *i* represents a logical set of the certain secret.

Figure 2-6 Defense Resource Allocation
Depend on the pattern of the shares and keys, network operators allocate defense resource to strengthen the nodal defense capability appropriately.

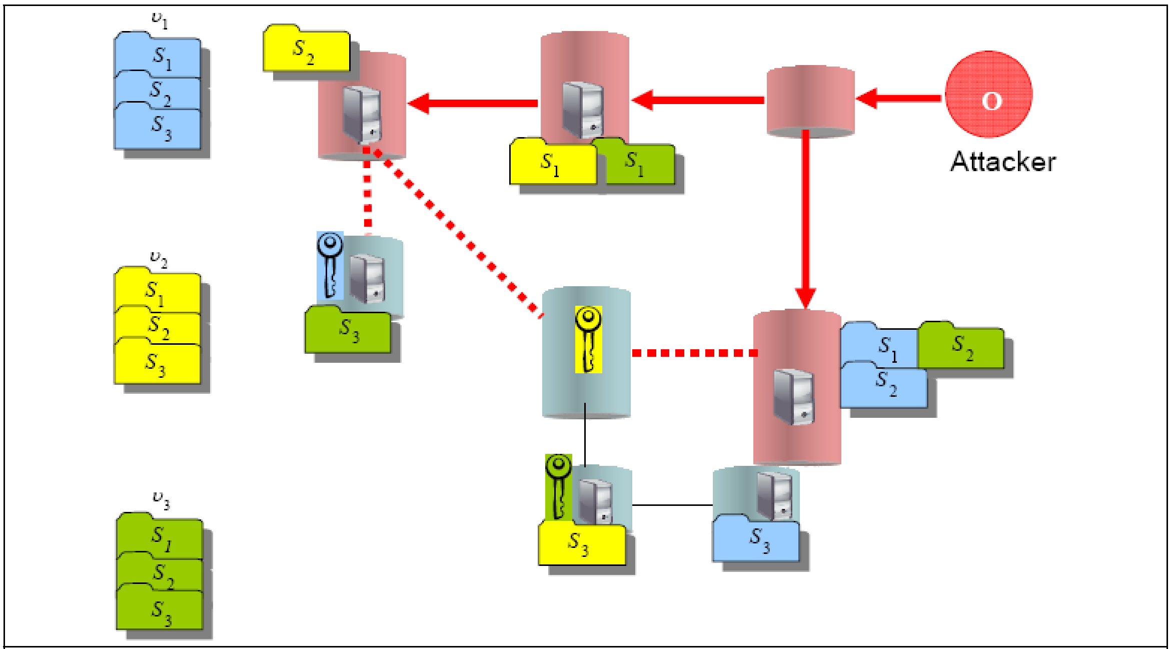


Figure 2-7 Attacking Targets
The attacker probes the neighbors of the current node to know how much power can compromise those nodes. They allocate their power to compromise nodes until all sensitive information revealed or the attack budget exhausted. No damage is caused in this case because all decrypt key is safe.

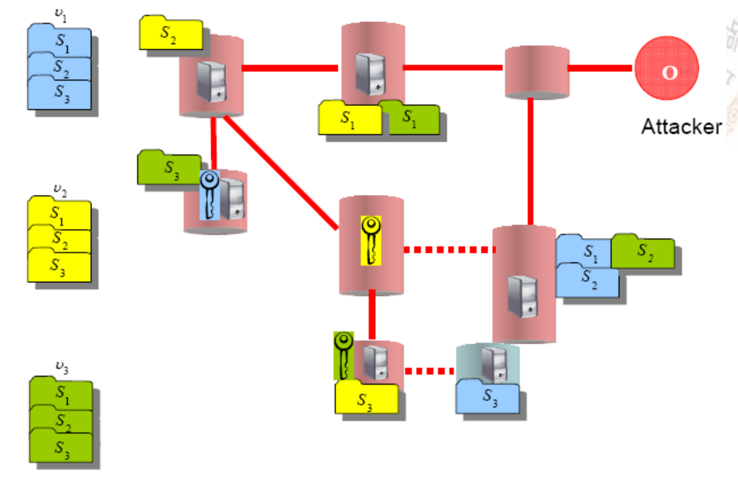


Figure 2-8 Attack Tree Construction
The attacker gets enough shares and corresponding keys to decrypt sensitive information causing leakage damage. The attack tree is constructed to reveal all sensitive information in this case.

○ Normal node
● Dummy node
Server
▨ Node defense capacity
🟥 Compromised node
Share
Key
——— Original link L_1
······ Artificial link L_2
······ Reachable link
━━━ Link on the attack tree
—▶ QoS routing path
⇨ Artificial flows

In this thesis, we model this scenario as the mathematical problem, called Network Planning Defense Strategy model (NPDS); and then arrange assumptions and descriptions of the NPDS model in Table 2-1.

Table 2-1 Assumptions and Description of the NPDS Model

Problem Assumptions:

1. The target network is at AS-level.
2. The network operator must construct the network topology with the viewpoint of network planning.
3. All nodes might be the candidate of the server and have transmission capability.
4. The artificial flows which are chosen from the artificial server to the legitimate user are disjoint paths in the terms of link.
5. A node is compromised if the attack resources applied to the node are equal to or more than the defense capability of the node.
6. Both the attacker and the network operator have complete information about the target network.
7. The objective of the attacker is to maximize the damage of information leakage by deciding which nodes to attack and allocating attack budget effectively.
8. The objective of the network operator is to minimize the damage and to satisfy QoS requirements simultaneously by means of choosing that the most robust network planning strategy and allocating defense budget appropriately.
9. Only node attacks are considered. (No link attacks are considered.)
10. Malicious attacks are considered.
11. The random errors of links are considered.
12. The sensitive information must be encrypted before processing the secret sharing.
13. The threshold number of secret sharing does not exceed the total pieces of the secret.
14. 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.
15. An attacker cannot recover the sensitive information unless he/she obtains at least the fixed number of shares and the corresponding decrypted key.

16. The network operator should make legitimate users get enough shares and decrypted key to recover the secret within the maximal tolerate time.
17. The attacker and the network operator have the budget limitation.

Given:

1. A set of feasible nodes and links in the AS
2. A set of dummy nodes
3. The distance between two nodes and the transmission latency of each link
4. The material type of the link
5. The implementation cost of each feasible link
6. The possibility of each link material occurs the random error
7. The set of all sensitive information
8. The damage incurred by information leakage
9. The maximal tolerate time of all the sensitive information
10. The defense capability function of each node
11. The total attack budget A
12. The total network operator budget B
13. Attacker's position O, which is connected to target network

Objective:

1. To minimize the maximized the damage of information leakage

Subject to:

1. Secret sharing constraints
2. Decrypted key constraints
3. Routing constraints
4. Delay constraints
5. Reliability constraints
6. The attack tree constraints
7. Attack budget A constraints
8. Defense budget B constraints

To determine:

1. Network operator:
   - Which link to set
   - What kind of the material type to choose for the link
   - Which nodes to put decrypted keys and shares

- The secret sharing strategy
- Which paths to transmit the shares and decrypted keys
- Which disjoint paths to guarantee reliability requirements
- The defense budget allocation strategy

2. Attacker:
   - Which sensitive information to obtain
   - Which nodes and paths to attack can maximize leakage damage

In Table 2-2, we show the given parameters in the NPDS model. For the attacker and the network operator, they know all given parameters, but there are few different given parameters between the NPDS model and the ATSS model. Next, we will use these parameters and variables to formulate the NPDS problem (IP 1).

Table 2-2 Given Parameters of the NPDS Model

Given parameters	
Notation	Description
N	The index set of all nodes, $N = N_1 \cup N_2$
N_1	The index set of all actual nodes
N_2	The index set of all dummy nodes which represent the whole secret on the node logically for the legitimate user
L	The index set of all links, $L = L_1 \cup L_2$
L_1	The index set of all candidate links
L_2	The index set of all artificial links which connected to the dummy node
W	The index set of all Origin-Destination (O-D) pairs for the attacker
p_w	The index set of all candidate paths for O-D pair w, where $w \in W$
R_{ij}	The index set of all candidate paths that the server *i* sends shares to the legitimate user *j*, where $i \in N, j \in N_1$
Φ_l	The latency of link l, where $l \in L_1$
C_i	The capacity of the node *i*, where $i \in N_1$
$\lambda_{j\upsilon}$	The indicator function, which is 1 if the legitimate user *j* requests the secret υ, and 0 otherwise (where $j \in N_1,\ \upsilon \in \nu$)
δ_{pl}	The indicator function, which is 1 if the link or the node l is on path p, and 0 otherwise (where $l \in (N_1 \cup L_1), p \in p_w$)
σ_{rl}	The indicator function, which is 1 if link l is on path r, and 0 otherwise (where $l \in L, r \in R_{ij}$)
ν	The index set of all sensitive information
S_υ	Damage incurred by leaking at least k_υ pieces of the secret υ and getting

	the corresponding decrypted key, where $\upsilon \in \nu$
m_υ	The share index set of the secret υ, where $\upsilon \in \nu$
M_υ	The size of the secret υ, where $\upsilon \in \nu$
T_υ	The maximal tolerable waiting time for legitimate users to use the secret υ, where $\upsilon \in \nu$
φ_{ij}	The number of the artificial flows between the server i and the user j, where $i \in N_2, j \in N_1$
A	The total attack budget of the attacker
B	The network planning budget of the network operator
ϑ	The index set of all material types of the links
β	The reliability requirement of the random error for legitimate users

In this formulation, the sensitive information is given the certain positive value S_υ, which the attacker tries to maximize the damage as possible. Therefore, his/her goal is collected the decrypted key and the shares of the same secret to reveal. The decision variables of NPDS problem are listed in Table 2-3.

Table 2-3 Decision Variables of the NPDS Model

Decision Variables	
Notation	Description
Ω_l	1 if link l is selected to implement, and 0 otherwise (where $l \in L_1$)
θ_l	The material type of the link l is chosen, where $l \in L_1$
$P_l(\theta_l)$	The probability function of the random error depends on its material, where $l \in L_1$
$\Gamma_l(\theta_l)$	The cost function depends on its material, where $l \in L_1$
$\eta_{i\upsilon}$	1 if the node i stored the decrypted key of the secret υ, and 0 otherwise (where $i \in N_1, \upsilon \in \nu$)
k_υ	The threshold number of shares required to recover the secret υ, where $\upsilon \in \nu$
$\alpha_{im\upsilon}$	1 if the node i stores the secret υ of the share of the index m, and 0 otherwise (where $i \in N_1, m \in m_\upsilon, \upsilon \in \nu$)
f_r	1 if the path r is selected to transmit the artificial flow, and 0 otherwise (where $r \in R_{ij}$)
V_{ijr}	1 if the path r is selected to transmit shares from the sever i to the legitimate user j and 0 otherwise (where $i, j \in N_1, r \in R_{ij}$)
b_i	The budget allocated to the node i to enhance the node's defense capability, where $i \in N_1$
$\hat{a}_i(b_i)$	The threshold of the attack cost leads to a successful attack, where $i \in N_1$

The NPDS problem is then formulated as the following problem (IP 1).

Objective function:

$$\underset{k_\upsilon,\Omega_l,b_i,\alpha_{im\upsilon},\eta_{i\upsilon}}{Min} \ , \ \underset{Z_\upsilon,y_i,a_i,x_p}{Max} \sum_{\upsilon\in\nu} S_\upsilon \cdot Z_\upsilon \qquad \text{(IP1)}$$

Subject to

$$\alpha_{im\upsilon} = 0 \; or \;\; 1 \qquad \forall i \in N_1, m \in m_\upsilon, \upsilon \in \nu \qquad \text{(IP1.1)}$$

$$\eta_{i\upsilon} = 0 \;\; or \;\; 1 \qquad \forall i \in N_1, \upsilon \in \nu \qquad \text{(IP1.2)}$$

$$|m_\upsilon| \ge k_\upsilon \qquad \forall \upsilon \in \nu \qquad \text{(IP1.3)}$$

$$\Omega_l = 0 \; or \;\; 1 \qquad \forall l \in L_1 \qquad \text{(IP1.4)}$$

$$\theta_l \in \vartheta \qquad \forall l \in L_1 \qquad \text{(IP1.5)}$$

$$\sum_{\upsilon\in\nu}\left(\frac{M_\upsilon}{|m_\upsilon|} \cdot \sum_{m\in m_\upsilon} \alpha_{im\upsilon}\right) \le C_i \qquad \forall i \in N_1 \qquad \text{(IP1.6)}$$

$$\sum_{i\in N_1} b_i + \sum_{l\in L_1} \Gamma_l(\theta_l) \cdot \Omega_l \le B \qquad \text{(IP1.7)}$$

$$V_{ijr} = 0 \; or \;\; 1 \qquad \forall i \in N_i, j \in N_1 \; r \in R_{ij} \qquad \text{(IP1.8)}$$

$$\sum_{r\in R_{ij}} (V_{ijr} \cdot \sigma_{rl}) \le \Omega_l \qquad \forall i, j \in N_1, l \in L_1 \qquad \text{(IP1.9)}$$

$$\sum_{i\in N_1} \sum_{m\in m_\upsilon} \alpha_{im\upsilon} \cdot V_{ijr} \ge k_\upsilon \cdot \lambda_{j\upsilon} \qquad \forall j \in N_1, \upsilon \in \nu, \; r \in R_{ij} \qquad \text{(IP1.10)}$$

$$\alpha_{im\upsilon} \cdot \sum_{l\in L_1} (\Phi_l \cdot V_{ijr} \cdot \sigma_{rl}) \cdot \lambda_{j\upsilon} \le T_\upsilon \qquad \forall i, j \in N_1, \upsilon \in \nu, \; m \in m_\upsilon, r \in R_{ij} \qquad \text{(IP1.11)}$$

$$\sum_{i\in N_1} \eta_{i\upsilon} \cdot V_{ijr} \ge \lambda_{j\upsilon} \qquad \forall j \in N_1, \upsilon \in \nu \; r \in R_{ij} \qquad \text{(IP1.12)}$$

$$\eta_{i\upsilon} \cdot \sum_{l\in L_1} (\Phi_l \cdot V_{ijr} \cdot \sigma_{rl}) \cdot \lambda_{j\upsilon} \le T_\upsilon \qquad \forall i, j \in N_1, \upsilon \in \nu, \; r \in R_{ij} \qquad \text{(IP1.13)}$$

$$f_r = 0 \; or \;\; 1 \qquad \forall r \in R_{ij}, i \in N_2, \; j \in N_1 \qquad \text{(IP1.14)}$$

$$\sum_{r\in R_{ij}} f_r = \varphi_{ij} \qquad \forall i \in N_2, j \in N_1 \qquad \text{(IP1.15)}$$

$$\sum_{i\in N_2} \sum_{r\in R_{ij}} f_r \cdot \sigma_{rl} \le \Omega_l \qquad \forall l \in L, j \in N_1 \qquad \text{(IP1.16)}$$

$$\sum_{l\in L_1} f_r \cdot \sigma_{rl} \cdot P_l(\theta_l) \le \beta \qquad \forall i \in N_2, j \in N_1, \; r \in R_{ij} \qquad \text{(IP1.17)}$$

$$\sum_{i\in N_1} a_i \le A \qquad \text{(IP1.18)}$$

$$0 \le a_i \le \hat{a}_i(b_i) \qquad \forall i \in N_1 \qquad \text{(IP1.19)}$$

$$\hat{a}_i(b_i) \cdot y_i \le a_i \qquad \forall i \in N_1 \qquad \text{(IP1.20)}$$

$$\sum_{p\in P_w} x_p = y_i \qquad \forall i \in N_1, w = (o,i) \qquad \text{(IP1.21)}$$

$$\sum_{p \in P_w} x_p \le 1 \qquad \forall w \in W \qquad \text{(IP1.22)}$$

$$x_p = 0 \ \ or \ \ 1 \qquad \forall p \in P_w, w \in W \qquad \text{(IP1.23)}$$

$$y_i = 0 \ \ or \ \ 1 \qquad \forall i \in N_1 \qquad \text{(IP1.24)}$$

$$\sum_{w \in W} \sum_{p \in P_w} x_p \cdot \delta_{pi} \le (|N_1| - 1) y_i \qquad \forall i \in N_1 \qquad \text{(IP1.25)}$$

$$\sum_{p \in P_w} x_p \cdot \delta_{pl} \le \Omega_l \qquad \forall l \in L_1, w \in W \qquad \text{(IP1.26)}$$

$$k_\upsilon \cdot Z_\upsilon \le \sum_{m \in m_\upsilon} \sum_{i \in N_1} (\alpha_{im\upsilon} \cdot y_i) \qquad \forall \upsilon \in \nu \qquad \text{(IP1.27)}$$

$$Z_\upsilon \le \sum_{i \in N_1} \eta_{i\upsilon} \cdot y_i \qquad \forall \upsilon \in \nu \qquad \text{(IP1.28)}$$

$$Z_\upsilon = 0 \, or \ \ 1 \qquad \forall \upsilon \in \nu. \qquad \text{(IP1.29)}$$

Explanation of the mathematical formulation:

- Objective function**:** The objective function is to minimize the maximized the information leakage damage $\sum_{\upsilon \in \nu} S_\upsilon \cdot Z_\upsilon$ caused by the attacker. In the inner problem, the attacker will adopt the effective strategy, which means the more beneficial targets to compromise, so as to recover more sensitive information. In addition, the attacker has to construct the attack path to get the decrypted key so that he/she can cause such damage. In the outer problem, the network operator constructs the most robust network topology with limited budget. It is essential for the network operator to determine appropriate recovered threshold k_υ, position to store shares $\alpha_{im\upsilon}$, the distribution of the decrypted keys $\eta_{i\upsilon}$, and defense resource allocation b_i on each node. By above strategies, the network operator tries to minimize the information leakage damage caused by the attacker.
- Constraint (IP 1.1) represents the node *i* contains the share *m* of the secret υ.
- Constraint (IP 1.2) represents the decrypted key of the secret υ on node *i*.
- Constraint (IP 1.3) restricts the threshold of the secret sharing can’t exceed the

total number of piece that the sensitive information divided into.

- Constraint (IP 1.4) enforces that the network operator decide whether to implement the link or not .The value of Ω_l is limited to 0 or 1.
- Constraint (IP 1.5) enforces the material of the link must be chosen from the set ϑ.
- Constraint (IP 1.6) represents the nodal capacity constraints. The total size of shares must not exceed the capacity of the node *i*.
- Constraint (IP 1.7) restricts the cost of network planning and defense resource allocation can't exceed the total budget B.
- Constraint (IP 1.8) and Constraint (IP 1.9) describe that the routing path *r* must route on the implemented link *l* from the source node *i* to the destination node *j*. The value of V_{ijr} is limited to 1 or 0.
- Constraint (IP 1.10) restricts that the legitimate user *j* must get the enough shares to recover the secret υ through all eligible paths of pair (*i*, *j*).
- Constraint (IP 1.11) represents that if the legitimate user *j* requests for the shares of the secret υ, the end to end delay of the chosen routing path *r* must be smaller than maximal tolerate time.
- Constraint (IP 1.12) restricts that the legitimate user *j* must get one decrypted key at least to recover the secret υ through all eligible paths of pair (*i*, *j*).
- Constraint (IP 1.13) represents that if the legitimate user *j* has request for the decrypted key of the secret υ, the end to end delay of the chosen routing path *r* must be smaller than maximal tolerate time.
- Constraint (IP 1.10) to Constraint (IP 1.13) jointly enforce that QoS routing requirements for all legitimate users in the network must be satisfied. The authenticate user *j* must get the threshold k_υ of shares and the corresponding

decrypted key through QoS routing to recover the secret within the maximal tolerate time T_{υ}.

- Constraint (IP 1.14) and Constraint (IP 1.15) enforce that the number of the end to end disjoint paths must be equal to the number of the artificial flows in order to satisfy the reliability of the network. The value of f_r is limited to 1 or 0.
- Constraint (IP 1.16) enforces that each link only is used by the artificial flow once at most for each user in order to construct the end to end disjoint paths from the dummy node to the legitimate user.
- Constraint (IP 1.17) restricts that the random error of each the disjoint path must be under the tolerable risk.
- Constraint (IP 1.18) and Constraint (IP 1.19) represent attack cost a_i applied to each node *i*. The total attack cost $\sum_{i\in N_I} a_i$ must not exceed the attack budget *A*. In addition, the attack cost a_i cannot exceed the defense capability of the node because of the waste of the attack cost.
- Constraint (IP 1.20) shows that the node is compromised successfully only if the attack cost applied to the node *i* being greater than its defense capability.
- Constraint (IP 1.21) represents that a node *i* is chosen for attack if and only if the attacker find a path between his initial position *o* and the targeted node *i*.
- Constraint (IP 1.22) restricts if the target node *i* is chosen, at most one attack path to reach the target node *i*.
- Constraint (IP 1.23) and Constraint (IP 1.24) are integer constraints, the value of x_p and y_i are 0 or 1.
- Constraint (IP 1.25) restricts that all actual nodes *i* on the attack path must necessarily be compromised. The attack path is transmits by the same node at

most $|N_1| - 1$ times in order to ensure no cycle on the attack tree.

- Constraint (IP 1.26) restricts the attack path p must construct on the implemented link.
- Constraint (IP 1.27) enforces that if the attacker wants to cause damage, he/she must get enough shares by means of compromising the nodes which contain the share of the secret υ.
- Constraint (IP 1.28) enforces that if the attacker wants to cause damage, he/she must get one decrypted key at least by means of compromising the nodes which contain the decrypted key of the secret υ.
- Constraint (IP 1.27) to Constraint (IP 1.29) jointly enforce that the attacker doesn't cause the information damage S_υ unless he gets decrypted key and reveals the threshold k_υ of shares by compromising the nodes.

2.3 Problem Formulation of the ATSS Model

To analyze the NPDS model, we should first solve the inner problem of the NPDS model, which is the ATSS model. The ATSS model represents the action attacker will adopt so as to cause the damage. Therefore, we must initiate some decision variables to given parameters in the ATSS model, such as Ω_l, $\alpha_{im\upsilon}$, $\eta_{i\upsilon}$, k_υ, and $\widehat{a}_i(b_i)$, which are the original decision variables in the NPDS model, become given parameters in the ATSS model.

Through solving this inner problem, the network operator can predict the behavior of the attacker, and he/she can adjust the strategy to reduce the damage. That is, while the ATSS problem is solved, the result is used as input into the NPDS model to find a better defense strategy against the attacker. The assumption and description are the same as the NPDS model. The given parameters and decision variables is shown in Table 2-4 and Table 2-5 respectively.

Table 2-4 Given Parameters of the ATSS Model

Given parameters	
Notation	Description
N_1	The index set of all actual nodes
L_1	The index set of all candidate links
W	The index set of all Origin-Destination (O-D) pairs for attack
p_w	The index set of all candidate paths for O-D pair w, where $w \in W$
δ_{pl}	The indicator function, which is 1 if the link or the node l is on path p, and 0 otherwise (where $l \in (N_1 \cup L_1), p \in p_w$)
Ω_l	1 if link l is selected to implement, and 0 otherwise (where $l \in L_1$)
ν	The index set of all sensitive information
m_υ	The share index of the secret υ, where $\upsilon \in \nu$
$\alpha_{im\upsilon}$	1 if the node i stores the share of the index m, and 0 otherwise (where $i \in N_1, m \in m_\upsilon, \upsilon \in \nu$)
$\eta_{i\upsilon}$	1 if the node i stored the decrypted key of the secret υ, and 0 otherwise (where $i \in N_1, \upsilon \in \nu$)
S_υ	Damage incurred by leaking at least k_υ pieces of the secret υ and getting the corresponding decrypted key, where $\upsilon \in \nu$

k_υ	The threshold number of shares required to recover the secret υ, where $\upsilon \in \nu$
$\hat{a}_i(b_i)$	The threshold of the attack cost leads to a successful attack, where $i \in N_1$
A	The total attack budget of attacker

Table 2-5 Decision Variables of the ATSS Model

Decision Variables	
Notation	Description
a_i	The attack budget allocated to compromise the node, where $i \in N_1$
Z_υ	1 if both k_υ shares and the key are stolen and 0 otherwise (where $\upsilon \in \nu$)
x_p	1 if path *p* is selected as the attack path; and 0 otherwise, where $p \in p_w$
y_i	1 if node *i* is attacked, and 0 otherwise (where $i \in N_1$)

We apply above given parameters and decision variables to formulate the ATSS model (IP 2) as the inner problem of NPDS model.

Objective function:

$$\underset{Z_\upsilon, y_i, a_i, x_p}{Max} \ (\sum_{\upsilon \in \nu} S_\upsilon \cdot Z_\upsilon) \qquad \text{(IP2)}$$

Subject to:

$$\sum_{p \in P_w} x_p = y_i \qquad \forall i \in N_1, w = (o, i) \qquad \text{(IP2.1)}$$

$$\sum_{p \in P_w} x_p \le 1 \qquad \forall w \in W \qquad \text{(IP2.2)}$$

$$x_p = 0 \ \ or \ \ 1 \qquad \forall p \in P_w, w \in W \qquad \text{(IP2.3)}$$

$$y_i = 0 \ \ or \ \ 1 \qquad \forall i \in N_1 \qquad \text{(IP2.4)}$$

$$\sum_{w \in W} \sum_{p \in P_w} x_p \cdot \delta_{pi} \le (|N_1| - 1) y_i \qquad \forall i \in N_1 \qquad \text{(IP2.5)}$$

$$\sum_{p \in P_w} x_p \cdot \delta_{pl} \le \Omega_l \qquad \forall l \in L_1, w \in W \qquad \text{(IP2.6)}$$

$$\sum_{i \in N_1} a_i \le A \qquad \text{(IP2.7)}$$

$$0 \le a_i \le \hat{a}_i(b_i) \qquad \forall i \in N_1 \qquad \text{(IP2.8)}$$

$$\hat{a}_i(b_i) \cdot y_i \le a_i \qquad \forall i \in N_1 \qquad \text{(IP2.9)}$$

$$k_\upsilon \cdot Z_\upsilon \le \sum_{m \in m_\upsilon} \sum_{i \in N_1} (\alpha_{im\upsilon} \cdot y_i) \qquad \forall \upsilon \in \nu \qquad \text{(IP2.10)}$$

$$Z_\upsilon \le \sum_{i \in N_1} \eta_{i\upsilon} \cdot y_i \qquad \forall \upsilon \in \nu \qquad \text{(IP2.11)}$$

$$Z_\upsilon = 0 \, or \ \ 1 \qquad \forall \upsilon \in \nu \qquad \text{(IP2.12)}$$

Explanation of the mathematical formulation:

- Objective function: The objective function of ATSS model is to maximize the damage, which causes by stealing enough shares and by getting corresponding decrypted keys to recover the secret. This problem also is the inner problem in the NPDS model. In addition, we transform (IP 2) from a maximization problem into an equivalent minimization problem for convenience. The transformation does not affect the substance of problem or finding optimal solution.
- Constraint (IP 2.1) to Constraint (IP 2.5) are the same as Constraints (IP 1.21) to Constraint (IP 1.25) in the NPDS model. The attacker chooses the attack path to reach the target *i*, and the intermediate nodes are all compromised.
- Constraint (IP 2.6) is the same as Constraint (IP 1.26), but the decision variable Ω_l in the NPDS model change into given parameters in the ATSS model.
- Constraint (IP 2.7) to Constraint (IP 2.9) are the same as Constraint (IP 1.18) to Constraint (IP 1.20). The attacker allocates his budget to compromise the node.
- Constraint (IP 2.10) is the same as Constraint (IP 1.27), but the decision variable k_υ in the NPDS model change into given parameters in the ATSS model.
- Constraint (IP 2.11) is the same as Constraint (IP 1.28), but the decision variable $\eta_{i\upsilon}$ in the NPDS model change into given parameters in the ATSS model.
- Constraint (IP 2.12) is the same as Constraint (IP 1.29).

Chapter 3 Solution Approach

3.1 Solution Approach for the ATSS Model

3.1.1 Lagrangean Relaxation Method

The optimization is one of the popular issues in the domain of science and plays an important role in application fields. For instance, nowadays, we apply the integer programming for some parameters to get the optimal solution in operation research. In fact, optimization techniques could be widely used to solve the computer networks. There are a lot of approaches to find optimal solutions of problems, and Lagrangean relaxation method was one of excellent optimization techniques. Lagrangean relaxation (LR), which is proposed in the 1973s, is general solution for large-scale mathematical programming problems [21][22], and use the concept is decomposition to exploit their special structure Its application contains linear programming, non-linear programming, integrity programming problem, etc. We illustrate the abstract concept of in Figure 3-1.

LR provides many significant merits [23] including that (i) it permits several ways to decompose the problem into the subproblems; (ii) in the subproblem, we can solve stand-alone problems optimally; (iii) we can get hints to obtain the boundary of the objective function value; (iv) we can design effective heuristic algorithms for solving the complex combinatorial problems.

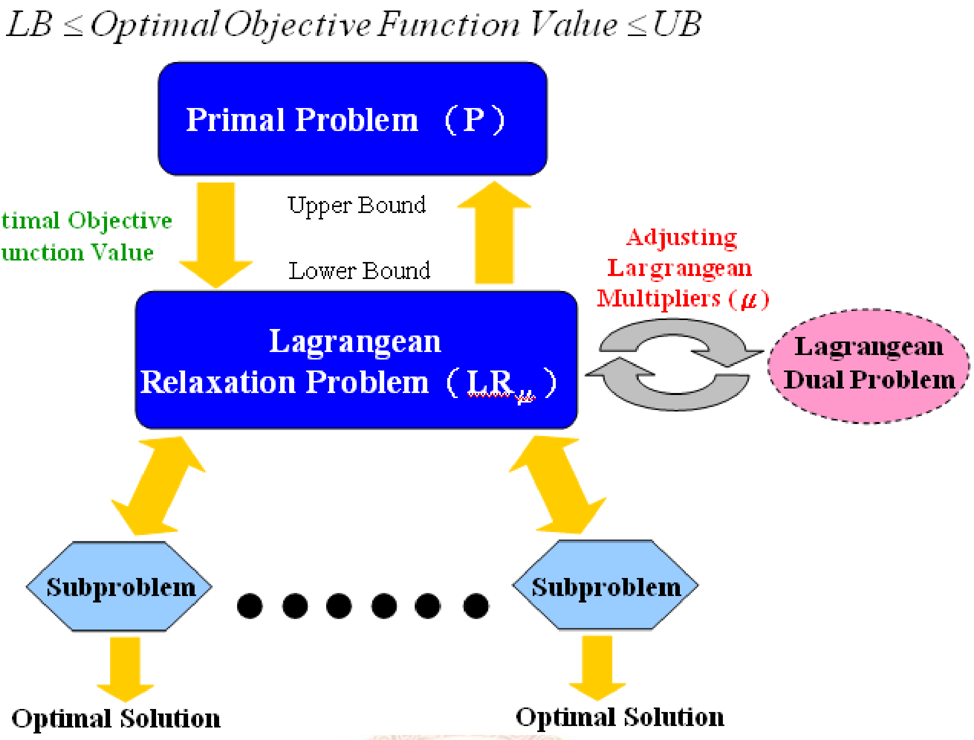

$LB \leq Optimal\,Objective\,Function\,Value \leq UB$
Primal Problem (P)
Optimal Objective Function Value
Upper Bound
Lower Bound
Adjusting Largrangean Multipliers (μ)
Lagrangean Relaxation Problem (LR_{μ})
Lagrangean Dual Problem
Subproblem
Subproblem
Optimal Solution
Optimal Solution

Figure 3-1 Concepts of the Lagrangean Relaxation Method

In this thesis, the attack-defense scenario is well-modeled as the mathematical optimization problem, which is too complicated to tackle in polynomial time. For this reason, we exploit the LR method to solve the mathematical problem. First, we must relax some complicating constraints, and the relaxed constraints multiply by the corresponding Lagrangean multipliers (μ) [23], and then add them to the primal objective function.

Second, we decompose Lagrangean relaxation problems into several subproblems according to the decision variables. To aim at each independent subproblem, we can propose the proper algorithm or adopt the well-known algorithm to solve it optimally and easily. By solving LR problem, we can get the lower bound (LB) of the primal objective function. If all decision variables are feasible and satisfy constraints from subproblems to the primal problem, the primal feasible solution is found, and then the outcome value is upper bound (UB). Otherwise, we must propose some heuristic algorithms to transform the infeasible decision variable into the feasible one. In order to get better solution quality, we use Lagrangean multiplier to adjust the original algorithm to a Lagrangean-based modified heuristic algorithm.

Third, we try to derive the tightest gap between the UB and LB, so we iteratively adjust the multipliers as better as possible, which is the so-called dual problem. In addition, to solve the Lagrangean dual problem, the subgradient optimization technique is usually applied.

Finally, the optimal objective function value in the primal problem is guaranteed within LB and UB. The detailed flow chart of Lagrangean relaxation method is shown in Figure 3-2. In following sections, we solved the ATSS problem by the Lagrangean relaxation method, and put the result of ATSS problem into the NPDS problem as the initial state. Through the Lagrangean relaxation procedure, we could get hints to tune decision variables in the NPDS model and executive it iteratively until the equilibrium of all decision variables.

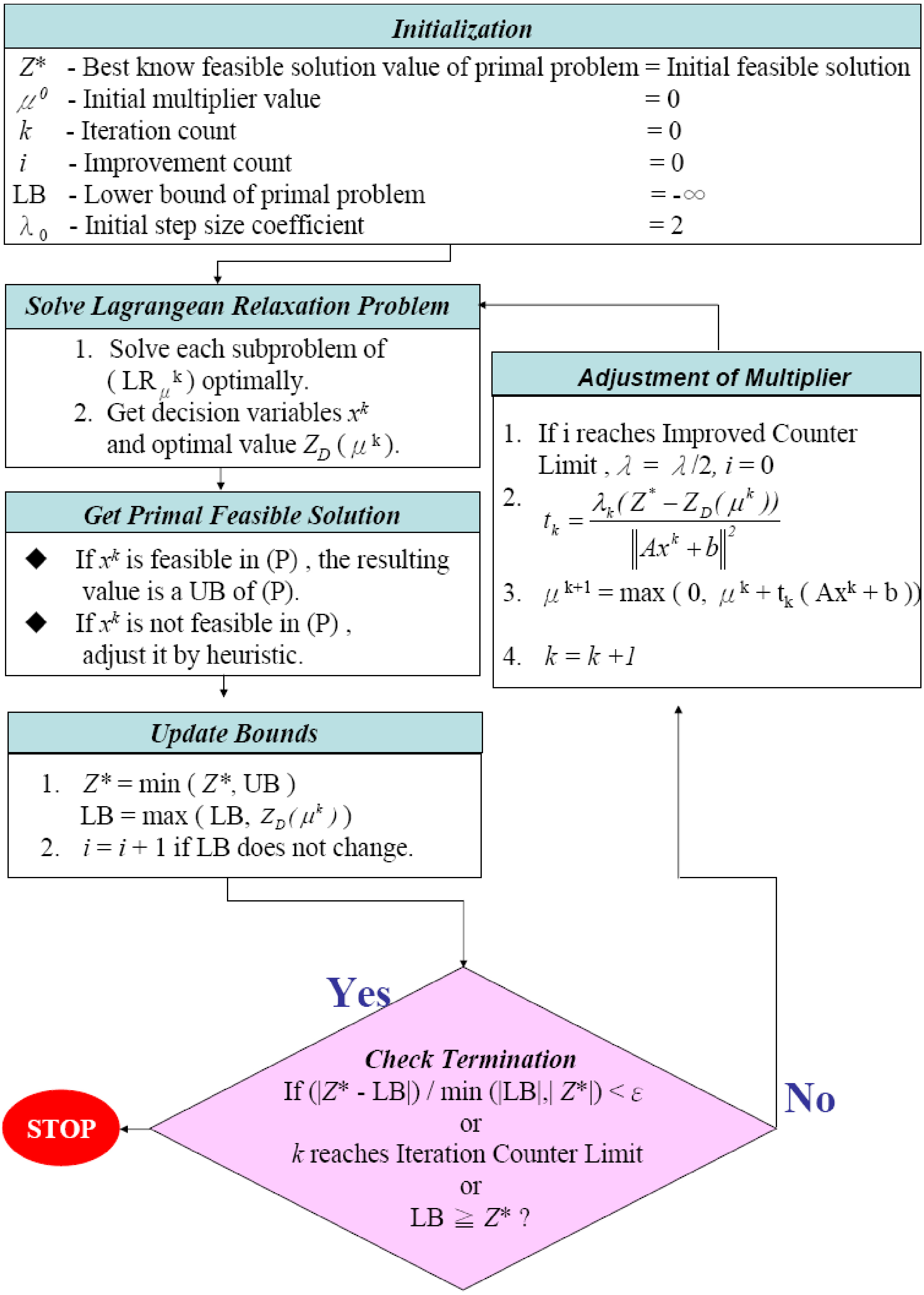

Initialization
Z^* - Best know feasible solution value of primal problem = Initial feasible solution
μ^0 - Initial multiplier value = 0
k - Iteration count = 0
i - Improvement count = 0
LB - Lower bound of primal problem = -∞
λ_0 - Initial step size coefficient = 2
Solve Lagrangean Relaxation Problem
1. Solve each subproblem of ($LR_{\mu}{}^{k}$) optimally.
2. Get decision variables x^k and optimal value $Z_D(\mu^k)$.
Get Primal Feasible Solution
◆ If x^k is feasible in (P) , the resulting value is a UB of (P).
◆ If x^k is not feasible in (P) , adjust it by heuristic.
Update Bounds
1. $Z^* = \min(Z^*, \text{UB})$
LB = max (LB, $Z_D(\mu^k)$)
2. $i = i + 1$ if LB does not change.
Adjustment of Multiplier
1. If i reaches Improved Counter Limit , $\lambda = \lambda/2$, $i = 0$
2. $t_k = \dfrac{\lambda_k(Z^* - Z_D(\mu^k))}{\left\|Ax^k + b\right\|^2}$
3. $\mu^{k+1} = \max(0, \mu^k + t_k(Ax^k + b))$
4. $k = k + 1$
Yes
Check Termination
If (|Z^* - LB|) / min (|LB|,| Z^*|) < ε
or
k reaches Iteration Counter Limit
or
LB ≧ Z^* ?
STOP
No

Figure 3-2 Lagrangean Relaxation Method Procedure

In order to solve the inner problem which is the ATSS model, we relax five constraints of (IP 2), and form the Lagrangean relaxation problem (LR 1).

3.1.2 Lagrangean Relaxation

We transform the primal problem into the Lagrangean Relaxation problem (LR 1) by means of the Lagrangean Relaxation method. The constraint (IP 2.1), (IP 2.5), (IP 2.9), (IP 2.10) and (IP 2.11) in the ATSS model are relaxed. Hence, we solve the optimization problem for (LR 1).

Optimization problem (LR 1):

$$\begin{aligned}
Z_D(\mu_1,\mu_2,\mu_3,\mu_4,\mu_5) = \min_{Z_\upsilon, y_i, a_i, x_p} & -\sum_{\upsilon\in\nu} S_\upsilon \cdot Z_\upsilon \\
& + \sum_{i\in N_1} \mu_i^1 \{ \sum_{p\in P(o,i)} x_p - y_i \} \\
& + \sum_{i\in N_1} \mu_i^2 \{ \sum_{w\in W} \sum_{p\in P_w} x_p \cdot \delta_{pi} - (|N_1| - 1) y_i \} \\
& + \sum_{i\in N_1} \mu_i^3 \{ \hat{a}_i(b_i) y_i - a_i \} \\
& + \sum_{\upsilon\in\nu} \mu_\upsilon^4 \cdot \{ k_\upsilon \cdot Z_\upsilon - \sum_{m\in m_\upsilon} \sum_{i\in N_1} \alpha_{im\upsilon} \cdot y_i \} \\
& + \sum_{\upsilon\in\nu} \mu_\upsilon^5 \{ Z_\upsilon - \sum_{i\in N_1} \eta_{i\upsilon} \cdot y_i \}
\end{aligned} \qquad \text{LR 1}$$

Subject to

$\sum_{p\in P_w} x_p \le 1$	$\forall w \in W$	LR 1.1
$x_p = 0 \text{ or } 1$	$\forall p \in P_w$	LR 1.2
$y_i = 0 \text{ or } 1$	$\forall i \in N_1$	LR 1.3
$\sum_{p\in P_w} x_p \cdot \delta_{pl} \le \Omega_l$	$\forall l \in L_1,$ $w \in W$	LR 1.4
$\sum_{i\in N_1} a_i \le A$		LR 1.5
$0 \le a_i \le \hat{a}_i(b_i)$	$\forall i \in N_1$	LR 1.6
$Z_\upsilon = 0 \text{ or } 1$	$\forall \upsilon \in \nu$	LR 1.7

Among Lagrangean multipliers $\mu_1, \mu_2, \mu_3, \mu_4$ and μ_5 are the vectors of $\{\mu_i^1\}, \{\mu_i^2\}$, $\{\mu_i^3\}, \{\mu_\upsilon^4\}, \{\mu_\upsilon^5\}$, which the multiplier μ_1 is unrestricted and $\mu_2, \mu_3, \mu_4, \mu_5$ are non-negative. Moreover, all multipliers are one-dimensional vectors. We decompose the LR problem into the following four independence subproblems and propose algorithms to solve them optimally.

Subproblem 1.1 (related to decision variable x_p)

$$Z_{Sub\,1.1}(\mu_1, \mu_2) = min\{\sum_{i \in N_1} \sum_{p \in P_{(o,i)}} \mu_i^1 \cdot x_p + \sum_{i \in N_1} \sum_{w \in W} \sum_{p \in P_w} \mu_i^2 \cdot x_p \cdot \delta_{pi}\} \qquad \text{Sub 1.1}$$

Subject to

$\sum_{p \in P_w} x_p \le 1$	$\forall w \in W$	Sub 1.1.1
$x_p = 0 \;\; or \;\; 1$	$\forall p \in P_w,$ $w \in W$	Sub 1.1.2
$\sum_{p \in P_w} x_p \cdot \delta_{pl} \le \Omega_l$	$\forall l \in L_1,$ $w \in W$	Sub 1.1.3

In this subproblem, each OD pair only can permit one path to be chosen from the constraint (Sub 1.1.1), and we can transform $\sum_{i \in N_1} \sum_{p \in P_{(o,i)}} \mu_i^1 \cdot x_p$ into $\sum_{w \in W} \sum_{p \in P_w} \mu_i^1 \cdot x_p + \sum_{p \in P_{(o,o)}} \mu_o^1 \cdot x_p$. Because no path has the same artificial link at the started node and the ended node, we can ignore $\sum_{p \in P_{(o,o)}} \mu_o^1 \cdot x_p$. Therefore, we further arrange the problem (Sub 1.1) to $\sum_{w \in W} \sum_{p \in P_w} [\sum_{j \in N_1} \mu_j^2 \cdot \delta_{pj} + \mu_i^1] \cdot x_p$, and decompose it into $|W|$ independent subproblems. We proposed the algorithm to solve each O-D pair *w= (o, i)* as following.

Step 1: We set μ_j^2 as the cost of the link weight as so to apply the Dijkstra's minimum cost shortest path algorithm. In addition, we must make sure that the link between nodes is already implemented on the path. The minimum cost path is calculated by the sum of the weight of the artificial links from the source to the destination for each O-D pair.

Step 2: For each O-D pair, we assign the other paths *p* to zero only one path being one, which is found by the shortest path algorithm in step 1. No more than one path can exist in the same O-D pair.

Step 3: For each chosen path from different O-D pairs, we examine the sum of the total cost on the path and the μ_i^1 value of its destination artificial link. If the value of the result is non-positive, we will set one to the corresponding x_p since this problem is the minimum problem. The value of the result is positive to assign x_p to zero.

The time complexity of Dijkstra's algorithm is O$(|N_1|^2)$. The source of the attacker is the same, so we just execute Dijkstra's algorithm one time. To sum up, the total complexity of the problem (Sub 1.1) is O$(|N_1|^2)$.

Subproblem 1.2 (related to decision variable Z_υ)

$$Z_{Sub\,1.2}(\mu_4, \mu_5) = min\{-\sum_{\upsilon \in v} S_\upsilon \cdot Z_\upsilon + \sum_{\upsilon \in v} \mu_\upsilon^4 \cdot k_\upsilon \cdot Z_\upsilon + \sum_{\upsilon \in v} \mu_\upsilon^5 \cdot Z_\upsilon\} \qquad \text{Sub 1.2}$$

Subject to

$$Z_\upsilon = 0 \; or \; 1 \qquad \forall \upsilon \in v \qquad \text{Sub 1.2.1}$$

In this problem (Sub 1.2), we arrange it to $\sum_{\upsilon \in v}(-S_\upsilon + \mu_\upsilon^4 \cdot k_\upsilon + \mu_\upsilon^5) \cdot Z_\upsilon$, and decompose into $|v|$ independent subproblems, where we decide the value of Z_υ of the secret υ. If $(-S_\upsilon + \mu_\upsilon^4 \cdot k_\upsilon + \mu_\upsilon^5)$ is non-positive, the value of Z_υ must be set to one for each sensitive information because of the minimum problem, and zero otherwise. In short, the rule is shown as following.

$$\begin{cases} 1, \ -S_{\upsilon} + \mu_{\upsilon}^{4} \cdot k_{\upsilon} + \mu_{\upsilon}^{5} \le 0 \\ 0, \ -S_{\upsilon} + \mu_{\upsilon}^{4} \cdot k_{\upsilon} + \mu_{\upsilon}^{5} > 0 \end{cases}$$

The time complexity of sub1.2 is $O(|v|)$.

Subproblem 1.3 (related to decision variable y_i)

$$Z_{Sub\,1.3}(\mu_1, \mu_2, \mu_3, \mu_4, \mu_5) = min \sum_{i \in N_1} \mu_i^{1}(-y_i)$$

$$+ \sum_{i \in N_1} - \mu_i^{2}(|N_1| - 1)y_i)$$

$$+ \sum_{i \in N_1} \mu_i^{3}(\hat{a}_i(b_i) \cdot y_i)$$

$$+ \sum_{\upsilon \in v} \sum_{m \in m_{\upsilon}} \sum_{i \in N_1} (-\mu_{\upsilon}^{4} \cdot \alpha_{im\upsilon} \cdot y_i)$$

$$+ \sum_{\upsilon \in v} \sum_{i \in N_1} - \mu_{\upsilon}^{5} \cdot \eta_{i\upsilon} \cdot y_i$$

Sub 1.3

Subject to

$y_i = 0 \; or \; 1$ $\quad \forall i \in N_1$ Sub 1.3.1

The same concept is presented above, accordingly the (Sub 1.3) reformed as $\sum_{i \in N_1} \{-\mu_i^{1} - \mu_i^{2}(|N_1| - 1) + \mu_i^{3} \cdot (\hat{a}_i(b_i)) + \sum_{\upsilon \in v}(\sum_{m \in m_{\upsilon}} (-\mu_{\upsilon}^{4} \cdot \alpha_{im\upsilon}) - \mu_{\upsilon}^{5} \cdot \eta_{i\upsilon})\} \cdot y_i$. Then we can further decompose into $|N_1|$ independent subproblems. We must determine the value of y_i of the actual node $i \in N_1$. Since the minimum problem, we set one if the sum of $(-\mu_i^{1} - \mu_i^{2}(|N_1| - 1) + \mu_i^{3} \cdot (\hat{a}_i(b_i)) + \sum_{\upsilon \in v}(\sum_{m \in m_{\upsilon}} (-\mu_{\upsilon}^{4}) \cdot \alpha_{im\upsilon} - \mu_{\upsilon}^{5} \cdot \eta_{i\upsilon})$ parameters are non-positive, zero otherwise. We apply the exhausting search algorithm to solve this subproblem. In short, the criterion is shown as following.

$$\begin{cases} 1, -\mu_i^1 - \mu_i^2(|N_1|-1) + \mu_i^3 \cdot (\widehat{a}_i(b_i)) + \sum\limits_{\upsilon \in V}(\sum\limits_{m \in m_\upsilon}(-\mu_\upsilon^4) \cdot \alpha_{im\upsilon} - \mu_\upsilon^5 \cdot \eta_{i\upsilon}) \le 0 \\ 0, -\mu_i^1 - \mu_i^2(|N_1|-1) + \mu_i^3 \cdot (\widehat{a}_i(b_i)) + \sum\limits_{\upsilon \in V}(\sum\limits_{m \in m_\upsilon}(-\mu_\upsilon^4) \cdot \alpha_{im\upsilon} - \mu_\upsilon^5 \cdot \eta_{i\upsilon}) > 0 \end{cases}$$

The time complexity of (Sub 1.3) is $O(|N_1| \cdot |\nu| \cdot |m_\upsilon|)$.

Subproblem 1.4 (related to decision variable a_i)

$$Z_{Sub\,1.4}(\mu_3) = min \sum_{i \in N_1} - \mu_i^3 \cdot a_i$$ Sub 1.4

Subject to

$$\sum_{i \in N_1} a_i \le A$$ Sub 1.4.1

$$0 \le a_i \le \widehat{a}_i(b_i) \qquad \forall i \in N_1$$ Sub 1.4.2

We can think of the problem (Sub 1.4) as a fractional knapsack problem, but something different is to minimize negative loss rather than to maximize positive profit traditionally. First we use the parameter $-\mu_i^3$ as the weight of the artificial link, and then we sort each actual node $i \in N_1$ by weight in ascending order. In addition, the value of the parameter μ_i^3 is non-negativity. Second, we allocate the value of a_i to $a_i(b_i)$ from the left until the sum of a_i exceeds *A*, or the next a_i is insufficient to set $a_i(b_i)$, which next a_i is set to remain value. Furthermore, the remainders are set to zero.

The time complexity of Sub (1.4) is $O(|N_1|^2)$

3.1.3 The Dual Problem and the Subgradient Method

In regard to the optimization problem (LR 1), it can be solved optimally after getting the optimal solutions of the four independence problems. According to the weak duality theorem, the objective value of $Z_D(\mu)$ is a lower bound (LB) of the primal problem provided for any multiplier $\mu \ge 0$ [21]. The dual problem (D 1) is used to calculate the tightest LB by adjusting the multipliers subject to $\mu \ge 0$.

Dual Problem (D 1)

$$Z_D = max\, Z_D(\mu_1, \mu_2, \mu_3, \mu_4, \mu_5) \qquad \text{(D 1)}$$

Subject to

$$\mu_2, \mu_3, \mu_4, \mu_5 \ge 0$$

The subgradient method is usually used to solve the dual problem. Denote the vector *m* be a subgradient of $Z_D(\mu_1, \mu_2, \mu_3, \mu_4, \mu_5)$. In iteration *k* of the subgradient optimization procedure, the multiplier vector is updated by $\mu^{k+1} = \mu^k + t^k m^k$, in which t^k is the step size determined by $t^k = \lambda \cdot \dfrac{Z^*_{IP2} - Z_D(\mu^k)}{\left\| m^k \right\|^2}$, where

$$m^k(\mu_1^k, \mu_2^k, \mu_3^k, \mu_4^k, \mu_5^k) = \Big(\sum_{p \in P(o,i)} x_p - y_i,\ \sum_{w \in W} \sum_{p \in P_w} x_p \delta_{pi} - (|N_1| - 1) y_i,$$
$$\hat{a}_i(b_i) y_i - a_i,\ k_\upsilon \cdot Z_\upsilon - \sum_{m \in m_\upsilon} \sum_{i \in N_1} \alpha_{im\upsilon} \cdot y_i,\ Z_\upsilon - \sum_{i \in N_1} \eta_{i\upsilon} \cdot y_i \Big)$$

Z^*_{IP2} is the upper bound (UB) of the primal objective function value (IP2) after iteration *k*, and λ is a scalar constant where $0 \le \lambda \le 2$. We must develop the algorithm to calculate the upper bound of the primal problem. In addition, the maximum number of the iterations in our proposed Lagrangean algorithm is 1000, and the improvement counter is 50. The constant λ in subgradient method initialized to be 2, which will be halved if the dual objective function value does not improve for 50 iterations.

3.1.4 Getting Primal Feasible Solutions

In order to improve the quality of the solution of Lagrangean Relaxation (LR) problem, we develop the Heuristic_LR_Algorithm to implement the procedure. In this method, we will adjust the solution from infeasible solution to feasible solution, which is getting from dual problem. The basic concept of the Heuristic_LR_Algorithm is that the attacker would first determine the certain secret as the target which he/she wants to recover while his/her budget is still sufficient. Furthermore, we would depend on the condition of the attacker's basket to set the damage value of each share or key.

If the node contains shares or keys of the secret υ, the node damage which is calculated by $\frac{\text{Secretdamage}[\upsilon]\cdot(1+\mu_{\upsilon}^{4}\cdot\mu_{\upsilon}^{5})}{\text{threshold}[\upsilon]\text{ - basket}[\upsilon]}$ must be multiplied by different coefficient value or make double secret damage in terms of the recovered condition of the secret in order to differentiate the importance of nodes. For example, we would strengthen the damage value as $\frac{\text{Secretdamage}[\upsilon]^{2}\cdot(1+\mu_{\upsilon}^{4}\cdot\mu_{\upsilon}^{5})}{\text{threshold}[\upsilon]\text{ - basket}[\upsilon]}$ if the secret is the recover-to-be. It's meaning that the closer to recover the secret is, the more damage we strengthen. In addition, we set node weight as $\frac{\hat{a}_i(b_i)^2}{Node[i].damage+(Node[i].damage/a_i)}$ because it could reflect the ratio of the attack cost to benefit gained.

Specially, we would recalculate the node weight continually when the attacker compromises new nodes and adds them to the attack tree. Moreover, we would set the weight of the compromised node to zero, so the attacker intends to choose compromised nodes to be the attack path because of reducing attack cost. Since the attacker must choose the secret as the target, we must calculate the sum of the path weight which connects with the component of the target.

Next, we compare to the sum of the path weight among all unrecovered secret, and the smallest the sum of path weight of the secret is set to the target. It is meaning that these paths to get the secret are the most profitable in the current attack scenario, and it is the best ratio of the attack cost to the benefit gained for the attacker. The procedure for choosing the to-be-recover target and constructing the attack path to unify the attack tree is repeated until the attacker has no attack power to compromise any other path. The detail procedure is shown as following in Table 3-2.

Table 3-2 Heuristic_LR_Algorithm

```
//Initialization
Initiate share_basket;
Initiate decrypted_key_basket;

For each attack-path p{
   If ( $X_p$  is assigned one in Sub_2.1){
      Add each Node i on attack-path p to attack-tree;
   }
}
For each secret  $\upsilon${
   Set    share_threshold[$\upsilon$]  =  $k_\upsilon$;
   Set    decrypted_key [$\upsilon$] = false;
}
//Calculate node damage
For each Node i on the attack-tree{
   For each secret  $\upsilon${
      If (Node i contains the share of the secret  $\upsilon$){
         Put this share into share_basket of the attacker;
          share_threshold[$\upsilon$]- -;
      }
      If (Node i contains the decrypted key of the secret  $\upsilon$){
         Put this key into decrypted_key_basket of the attacker;
         decrypted_key [$\upsilon$] = true;
      }
   }
}
Calculate attack_total_cost of attack-tree;
//Check the condition of the secret and set the damage of the node dynamically
according to the attacker's basket;
If (attack_total_cost <ATTACK_BUDGET){
   While(attack_total_cost<ATTACK_BUDGET AND some secrets are unchecked){
   //Calculate the weight of the node dynamically
      For each Node i{
         Node[i].damage = 0;
         If(Node i is compromised){
```

```
        $Node[i].weight = 0$;
    }
    Else{    //Only set weight to uncompromised nodes
        For each Secret $\upsilon$ {
            If(Secret $\upsilon$ is not recovered){
                //( share_threshold[$\upsilon$]<= 0 ) AND (!$decrypted\_key\,[\upsilon]$ )
                If(Node $i$ contains the decrypted key that the attacker needs){
                    $Node[i].damage += Secretdamage[\upsilon]^2 \cdot (1+\mu_\upsilon^4 \cdot \mu_\upsilon^5)$
                }
                //( share_threshold[$\upsilon$]> 0 ) AND ($decrypted\_key\,[\upsilon]$ )
                If(Node $i$ contains the share that the attacker needs){
                    $Node[i].damage += \frac{Secretdamage[\upsilon]^2 \cdot (1+\mu_\upsilon^4 \cdot \mu_\upsilon^5)}{threshold[\upsilon] - basket[\upsilon]}$
                }
                //( share_threshold[$\upsilon$]> 0 ) AND (!$decrypted\_key\,[\upsilon]$ )
                If(Node $i$ contains the share that the attacker doesn’t need){
                    $Node[i].damage += \frac{Secretdamage[\upsilon] \cdot (1+\mu_\upsilon^4 \cdot \mu_\upsilon^5)}{2 \cdot (threshold[\upsilon] - basket[\upsilon])}$
                }
                //( share_threshold[$\upsilon$]> 0 ) AND (!$decrypted\_key\,[\upsilon]$ )
                If(Node $i$ contains the decrypted key that the attacker doesn’t need){
                    $Node[i].damage += \frac{Secretdamage[\upsilon] \cdot (1+\mu_\upsilon^4 \cdot \mu_\upsilon^5)}{(threshold[\upsilon] - basket[\upsilon])}$
                }
            }
        }
        If ( $a_i > 0$)
            $Node[i].weight = \frac{\hat{a}_i(b_i)^2}{Node[i].damage + (Node[i].damage/a_i)}$;
        Else
            $Node[i].weight = \frac{\hat{a}_i(b_i)^2}{Node[i].damage}$;
    }
}
//Choose the secret that the attacker decides to recover it
$Target\_Secret$ =$Find\_\,Target\_Secret()$; // shown as Table 3.5
For each first $k_\upsilon th$ Node $i$ contains the component of $Target\_Secret$ {
    IF($attack\_total\_cost + path\_cost$ of Node $i$ <= $ATTACK\_BUDGET$){
        Compromise Node $i$ and all nodes on the chosen path;
        Add these nodes to $attack\text{-}tree$;
        $Attack\_total\_cost$ += $path\_cost$ of Node $i$;
    }
}
For each secret $\upsilon$ {
    If(Node $i$ contains shares of secret $\upsilon$ )
```

```
            share_threshold[$\upsilon$]- -;
          If(Node *i* contains key of secret $\upsilon$ )
            $decrypted\_key[\upsilon] = true$;
        }
        Update *share_basket*;
        Update *decrypted_key_basket*;
    }
}
else{
    While(*attack_total_cost* > *ATTACK_BUDGET*){
        //Calculate the weight of the leaf Node dynamically
        For each leaf Node *i*{
          For each secret $\upsilon${
            If(Node *i* contains secret $\upsilon$){
              If( $share\_threshold[\upsilon]$ <= 0 AND $decrypted\_key[\upsilon]$ )
                $Node[i].damage \mathrel{+}= Secretdamage[\upsilon]^{2} \cdot (1+\mu_{\upsilon}^{4} \cdot \mu_{\upsilon}^{5})$;
              If( ($decrypted\_key[\upsilon]$   AND   $share\_threshold[\upsilon]$ > 0)
               OR (!$decrypted\_key[\upsilon]$   AND   $share\_threshold[\upsilon]$ <= 0) )
                $Node[i].damage \mathrel{+}= Secretdamage[\upsilon] \cdot (1+\mu_{\upsilon}^{4} \cdot \mu_{\upsilon}^{5})$;
              If(!$decrypted\_key[\upsilon]$   AND   $share\_threshold[\upsilon]$ > 0)
               $Node[i].damage \mathrel{+}= \frac{Secretdamage[\upsilon] \cdot (1+\mu_{\upsilon}^{4} \cdot \mu_{\upsilon}^{5})}{2}$;
            }
          }
          If ($a_i$ > 0)
            $Node[i].weight = \frac{\hat{a}_i(b_i)^2}{Node[i].damage + (Node[i].damage/a_i)}$;
          Else
            $Node[i].weight = \frac{\hat{a}_i(b_i)^2}{Node[i].damage}$;
        }
        Sort nodes which are leaf_node by weight in ascending order;
        Choose Node *i* with the largest weight node among leaf-nodes;
        Remove Node *i* from *attack-tree*;
        *Attack_total_cost* -= *attack_cost* of Node *i*;
        For each secret $\upsilon${
          If(Node *i* contains shares of secret $\upsilon$)
             share_threshold[$\upsilon$]++;
          If(Node *i* contains decrypted key of secret $\upsilon$)
            $decrypted\_key[\upsilon] = false$;
        Update *share_basket*;
        Update *decrypted_key_basket*;
        }
    }
}
```

Table 3-3 Find_ Target_Secret Algorithm

```
//Initialization
Use Node weight as the cost to implement *Prim's* Alogrithm;
//Calculate how difficult to recover the each Secret
For each Secret $\upsilon$ {
    Find those nodes which contains the shares of the secret $\upsilon$ and mark them;
    Calculate the shortest paths to these marked nodes;
    For each path which is first $k_\upsilon$ *th* weight paths{
        *Secret_weight[$\upsilon$] += the path weight of Share_weight[$\upsilon$][k]*
    }
    Find the smallest path weight of the node which contains the decrypted key;
    *Secret_weight[$\upsilon$] += the path weight of key_weight[$\upsilon$]*
}
Sort each *Secret_weight[$\upsilon$]*;
Find the smallest *Secret_weight[$\upsilon$]* and mark it as the *target_secret*;
Mark this secret $\upsilon$ as the checked secret;
```

3.1.5 Summary of the Solution Approach for the ATSS Model

We use Lagrangean Relaxation-based algorithm to solve ATSS problem model and denote it as the LR. What's more, the relaxed subproblems all are solved optimally, and the result is denoted as Z_d also LB, then we can obtain Z_{IP} also UB the from the heuristic algorithm we proposed. In order to narrow the gap between LB and UB, the LR procedure is repeated iteratively to adjust Lagrangean multipliers until the stop condition is met. As shown in Table 3-4, it goes into a more detail about the complete LR algorithm for solving ATSS Model.

Table 3-4 Lagrangean Relaxation Algorithm

```
//Objective: maximize the total reveal secret damage, $min\,(-Z_{IP2})$
//Initialization of multipliers
Initialize the Lagrangean multiplier vectors
$(\mu_1, \mu_2, \mu_3, \mu_4, \mu_5)$ and all to be zero vectors;
*UB = 0*; *LB = -TOTAL_REVEAL_OF_SECRET*; *//LB =* $-\sum_{\upsilon\in V} S_\upsilon \cdot Z_\upsilon$
*Improvement_counter = 0*;
$\lambda = 2$;
*ITERATION_COUNTER_LIMIT = 1000*;
*Init_Defense_Strategy();*
```

```
FOR iteration = 1 TO ITERATION_COUNTER_LIMIT {
    Solve (Sub 1.1);
    Solve (Sub 1.2);
    Solve (Sub 1.3);
    Solve (Sub 1.4);
    Calculate $Z_D$;
    $Z^*_{IP2}$ = –Heuristic_LR();
//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();
}
```

3.2 Solution Approach for the NPDS Model

In the ATSS model, we first deploy the initial the network environment which is satisfied the QoS constraints for legitimate users. The result of the ATSS model indicates the best strategy the attacker adopts under the certain the network environment. Next, the optimal outcome of the ATSS model is used as the input of the NPDS model, whose objective function is not only to minimize the information damage revealed by the attacker but also to achieve the QoS constraints. Therefore, we must determine the best location the shares and decrypt keys stored and what kind of threshold being better, and adjust the budget allocation strategy according to the LR-based algorithm in the ATSS model. After the adjustment procedure, we put the outcome of the NPDS model into the LR procedure again. Figure 3.3 illustrates the main concepts of the solution approach for the NPDS model and the detailed flow chart is presented.

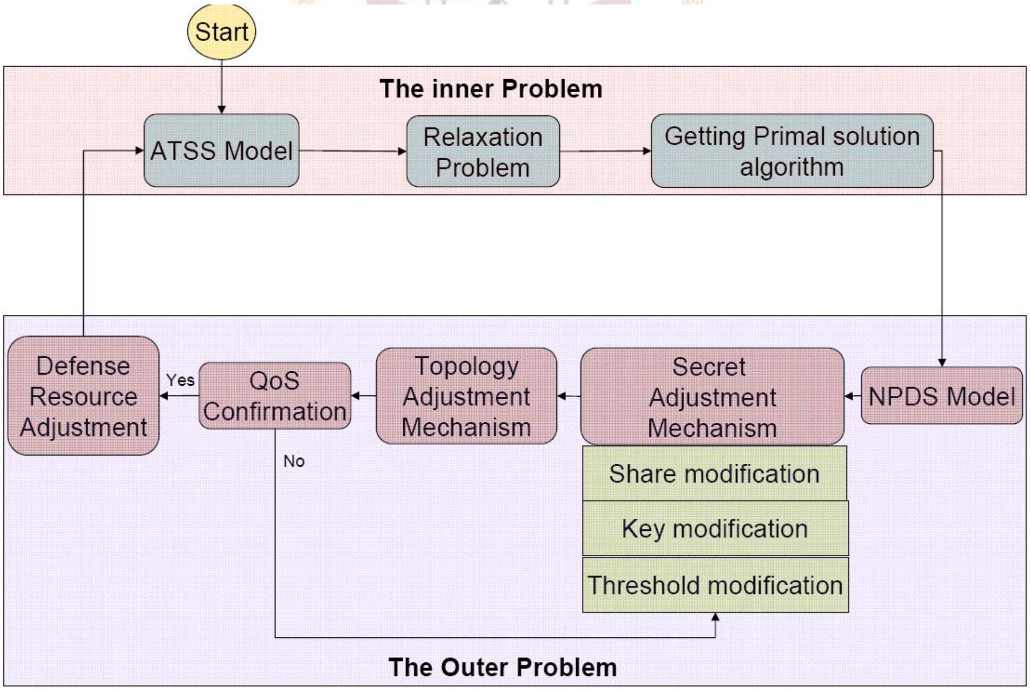


Figure 3-3 Solution Approach for the NPDS Model

The adjustment procedure of the NPDS model will change continually until the condition between the attacker strategy and defense strategy balanced. The adjustment procedure mainly divided into three sections: Secret Adjustment Mechanism, Topology adjustment Mechanism and Defense Resource Adjustment Mechanism.

In the first mechanism, we calculate how many times the secret is recovered and try to move the component (including shares or the decrypted key) of the most times recovered secret from compromised nodes to uncompromised nodes for the sake of finding the appropriate location to store the secret securely. Furthermore, we also can enhance the threshold of the recovered secret to reduce the damage caused by attackers.

In the second mechanism, we depend on the new deployed pattern to reconstruct the network topology, and we will first rebuild the links among the most times hop sites by the attacker. The improvement ratio λ determines how many nodes the network operator tries to reconstruct links. If the damage is still not improved by iterations, it means the variation is too significant so that we must adjust λ to solve the problem again. Before executing the third mechanism, the QoS constraints for legitimate users must be satisfied. For the system reliability problem which is the random error of link considered, we design the artificial capacity of the link as one and artificial flows for OD pairs, and then we apply min cost flow algorithm to find the link disjoint paths. In addition, we must make sure the reliability of each link disjoint path over the system reliability requirement. If all QoS constraints for each user are guaranteed, the Defense Resource Adjustment Mechanism will process; otherwise, we must redeploy the pattern of the secret.

The concept of the third mechanism is the same as the proverb “Spend every worth penny.” At first, we check the state of each node in the network. If the node is uncompromised, it means that there is unnecessary defense budget allocated on this node. Either way, the compromised nodes must be allocated more defense budget than the current state. For this reason, we define the extraction ratio θ to extract the defense budget of uncompromised nodes to compromised nodes. Besides, the extraction is equal to the step size coefficient. This mechanism does not improve within a certain number of iterations because it may extract too much budget from uncompromised nodes. Consequently, we halve the step size coefficient and consider nodes whether they are the hop sites for the attacker to determine how much ratio we can extract. The more times the node is used as the attack tree, the more important the node represents.

The whole heuristic algorithm for the NPDS model is shown in Table 3-5, called Heuristic_NPDS, and then we presented the core algorithm of the Secret_Adjustment, Topology_Adjustment, and Defense_Resource_Adjustment below.

Table 3-5 The Heuristic_NPDS Algorithm

```
//Objective: minimize the maximized total damage. min (max  $Z_{IP1}$ )
//Initialization
Share_and_key_deployment();
Verify QoS constraints();
Init_Defense_Strategy();
UB= -LR();    //the return value of LR() is negative due to the objective function
                   transformation in the ATSS model
Improvement_counter=0;
Improvement_topology_counter=0;
$\theta = 0.5$;
$\lambda = 0.25$;
//Main Heuristic_NPDS procedure
For iteration=1 To ITERATION_COUNTER_LIMIT{
   If(iteration < ITERATION_COUNTER_LIMIT/2){
      Secret_adjustment();                    //as shown in Table 3.6
      Topology_ adjustment ($\lambda$);              //as shown in Table 3.8
      Improvement_topology_counter ++;
   }
   Else{
      Defense_resource_adjustment ($\theta$);    //as shown in Table 3.9
      Improvement_counter ++;
   }
   $Z^*_{IP1}$ = -LR();
   If( $Z^*_{IP1} < UB$){
      UB= $Z^*_{IP1}$;
      Improvement_topology_counter=0;
      Improvement_counter =0;
```

```
    }
    //Update step size
    If(Improvement_topology_counter >= Improvement_topology_LIMIT){
        $\lambda = \lambda \cdot 0.8$;
        Improvement_ topology_counter=0;
    }
    If(Improvement_counter >= Improvement_counter _LIMIT){
        $\theta = \theta / 2$;
        Improvement_counter =0;
    }
}
```

Table 3-6 The Secret_Adjustment Algorithm

```
//Initialization
//For each recovered secret, to move the key or share to more secure location,
otherwise enhance its threshold under the QoS constraint
For each recovered secret $\upsilon$ by the attacker{
    Find the node $i$ which stores the decrypted key of the secret $\upsilon$ ;
    Find the node $k$ which is the uncompromised node and its weight is
    larger than node $i$;
    If(node $k$ could be found AND it doesn’t affect the QoS constraints for legitimate
      users){
        //Exchange the location of key and share
        move the decrypted key of the secret $\upsilon$ to node $k$;
        move the share of the secret $\upsilon$ to node $i$;
        Share_Reallocation ($\upsilon$);     //as shown in Table 3-7
    }
    Else{
        push the secret $\upsilon$ into the collection_vector;
    }
}
While(collection_vector is not empty){
    Enhance the threshold of the secret $\upsilon$ in collection_vector;
    //To avoid that the secret can’t be recovered by users in the tolerate time
    If(the action affect the QoS constraints){
        Execute replication mechanism for the unsatisfied users;
    }
    Remove the secret $\upsilon$ from collection_vector;
}
```

Table 3-7 The Share_Reallocation Algorithm

```
//Initialization
For each compromised node $i${
    if(Node $i$ contains shares of the secret $\upsilon$){
        Remove shares of the secret $\upsilon$ from Node $i$ then put them into Collection;
    }
}
```

```
Update each Node weight;
//Move shares to more the secure location
For each uncompromised Node i whose weight from the largest to the smallest{
   If(Node i satisfied QoS constraints AND its capacity is available){
      Allocate one share to Node i;
      Collection--;
   }
}
If(Collection is not empty){    //Reallocate remaining shares
   For each compromised Node i whose weight from the largest to the smallest{
      If(Node i satisfied QoS constraints AND its capacity is available){
         Allocate one share to Node i;
         Collection--;
      }
      If(Collection is not empty)
         Break;
   }
}
```

Table 3-8 Topology_ Adjustment

```
//Initialization
//try to change hop-site
Modify_node_number = $\lambda \cdot$ Compromised_nodes;
Find the first Modify_node_number- th times used nodes AND put them into
Candidate_vector;
//try to enhance its depth defense capability
For each node i in Candidate_vector{
   For each link l of the node i {
      If(the link l is used to construct the attack path){
         If (min cost flow algorithm is still satisfied for all legitimate users without
           the link l){
            Remove the link l;
            Basket += the cost of link l;
         }
         Else{
            //the link l could not be remove directly then to find the alternative link $l_1$
            //node k is the other node which the link l connected
            //try to find the new link connected with node k and node h
            If(there is an uncompromised node h which is the largest weight
              AND the new link still satisfied min cost flow algorithm for all
              legitimate users AND the cost of link $l_1 \le$ (the cost of link l + Basket)){
               Destroy the link l between node k and node i;
               Construct the new link $l_1$ between node k and node h;
            }
         }
      }
   }
}
```

Table 3-9 Defense_Resource_Adjustment

```
//Initialization
Defense_cost = 0;
For each uncompromised nodes i in the network{
//Take unnecessary budget to basket depending on its importance; $w_i$ is the number of times used of the node i by the attacker; $b_i$ is the defense budget.
    $b_i = b_i(1 - \theta(1 - \frac{w_i}{w_{max}}));$
    Defense_cost + = $b_i$;
}
Basket += Total_Defense_budget - Defense_cost;
//Reallocate more defense budget to compromised nodes
For each compromised nodes i in the network{
    $b_i$ += Basket * Budget_Reallocation();
}
```

3.3 The Independent Model – the DDS Model

In this section, we further formulate the independent single layer model from the NPDS problem, called Discrete Degree of Secret model (DDS) as our second solution approach. In other words, we would obtain the better initial network deployment with this model, and the constraints of DDS model are the same as the constraints of NPDS model. However, there is no the existence of attacker in the DDS model, so we must define the metric “Discrete Degree” as the objective function, denoted by (IP 3) in order to represent the impact of the attacker under the initial network deployment. Here, we first solve the DDS model and then put the result into the ATSS model. Hence, we consider the model as the second solution approach in our research.

3.3.1 Independent Problem Description and Formulation

In the DDS model, we not only satisfy the QoS requirements but also try to optimize the discrete degree of secrets. The attacker must make more effort to recover the secret if the discrete degree of secrets is superior. Furthermore, we choose the best 10 solutions as the input of the ATSS model to evaluate the impact of the attacker. Most of the notations used in the formulation are the same as the NPDS model; only one additional decision variable is induced to represent the discrete degree of secrets and listed in Table 3-10.

Table 3-10 Additional Decision Variable in (IP 3)

Decision Variable	
Notation	Description
$\omega_{\mathrm{i}\upsilon}$	1 if node *i* contains the shares or decrypted keys of the secret υ, and 0 otherwise (where $i \in N_1, \upsilon \in \nu$)

Objective function:

$$\underset{k_\upsilon, \Omega_l, \alpha_{im\upsilon}, \eta_{i\upsilon}}{Min} \{ \sum_{i \in N_1} \frac{|\nu|}{\sum_{\upsilon \in \nu} \omega_{i\upsilon}} + \sum_{i=0}^{|N_1|-1} \sum_{j=i+1}^{|N_1|} \frac{(\sum_{\upsilon \in \nu} \omega_{i\upsilon} \cdot \omega_{j\upsilon})^2}{|\nu|} \} \qquad \text{(IP3)}$$

Subject to

$$\alpha_{im\upsilon} = 0 \; or \;\; 1 \qquad \forall i \in N_1, m \in m_\upsilon, \upsilon \in \nu \qquad \text{(IP3.1)}$$

$$\eta_{i\upsilon} = 0 \;\; or \;\; 1 \qquad \forall i \in N_1, \upsilon \in \nu \qquad \text{(IP3.2)}$$

$$\omega_{i\upsilon} = 0 \; or \; 1 \qquad \forall i \in N_1, \upsilon \in \nu \qquad \text{(IP3.3)}$$

$$\alpha_{im\upsilon} \le \omega_{i\upsilon} \qquad \forall i \in N_1, m \in m_\upsilon, \upsilon \in \nu \qquad \text{(IP3.4)}$$

$$\eta_{i\upsilon} \le \omega_{i\upsilon} \qquad \forall i \in N_1, \upsilon \in \nu \qquad \text{(IP3.5)}$$

$$|m_\upsilon| \ge k_\upsilon \qquad \forall \upsilon \in \nu \qquad \text{(IP3.6)}$$

$$\sum_{\upsilon \in \nu} (\frac{M_\upsilon}{|m_\upsilon|} \cdot \sum_{m \in m_\upsilon} \alpha_{im\upsilon}) \le C_i \qquad \forall i \in N_1 \qquad \text{(IP3.7)}$$

$$V_{ijr} = 0 \; or \;\; 1 \qquad \forall i \in N_i, j \in N_1 \; r \in R_{ij} \qquad \text{(IP3.8)}$$

$$\sum_{r \in R_{ij}} (V_{ijr} \cdot \sigma_{rl}) \le \Omega_l \qquad \forall i, j \in N_1, l \in L_1 \qquad \text{(IP3.9)}$$

$$\sum_{i \in N_1} \sum_{m \in m_\upsilon} \alpha_{im\upsilon} \cdot V_{ijr} \ge k_\upsilon \cdot \lambda_{j\upsilon} \qquad \forall j \in N_1, \upsilon \in \nu, r \in R_{ij} \qquad \text{(IP3.10)}$$

$$\alpha_{im\upsilon} \cdot \sum_{l \in L_1} (\Phi_l \cdot V_{ijr} \cdot \sigma_{rl}) \cdot \lambda_{j\upsilon} \le T_\upsilon \qquad \forall i, j \in N_1, \upsilon \in \nu, m \in m_\upsilon, r \in R_{ij} \qquad \text{(IP3.11)}$$

$$\sum_{i \in N_1} \eta_{i\upsilon} \cdot V_{ijr} \ge \lambda_{j\upsilon} \qquad \forall j \in N_1, \upsilon \in \nu \; r \in R_{ij} \qquad \text{(IP3.12)}$$

$$\eta_{i\upsilon} \cdot \sum_{l \in L_1} (\Phi_l \cdot V_{ijr} \cdot \sigma_{rl}) \cdot \lambda_{j\upsilon} \le T_\upsilon \qquad \forall i, j \in N_1, \upsilon \in \nu, r \in R_{ij} \qquad \text{(IP3.13)}$$

Explanation of the mathematical formulation:

- Objective function: The objective is to minimize the metric “Discrete Degree” which is used to evaluate the efficiency of the attacker under the specific QoS constraints. The former equation $\sum_{i\in N_I}\frac{|V|}{\sum_{\upsilon\in V}\omega_{i\upsilon}}$ describes the separation of shares in terms of the single secret, and the later equation $\sum_{i=0}^{|N_I|-1}\sum_{j=i+1}^{|N_I|}\frac{(\sum_{\upsilon\in V}\omega_{i\upsilon}\cdot\omega_{j\upsilon})^2}{|V|}$ is used to compare share patterns among all nodes; the numerator is squared to emphasis the importance of different share patterns of each node. The smaller the value of the objective function is, the less probability of secrets recovered by the attacker is.
- Constraint (IP 3.3) to Constraint (IP 3.5) describe the decision variable $\omega_{i\upsilon}$ is set one if only if the node i contains the shares or decrypted keys of the secret υ.
- Other constraints are all the same as constraints of the NPDS model.

3.3.2 The Solution Approach for the DDS Model

The solution approach of the DDS model is proposed by Simulated Annealing method [24]. The Simulated Annealing (SA) method is the iterative improvement approach to solve the combinatorial optimization problem. The basic concept of SA is to simulate the natural material cooling and crystallizing steady state. In the annealing processing, the atoms of the material are unsteady and unstuck state under the high temperature because of high energy; in the cooling processing, the atoms would be tight and crystal with the drop of the temperature. In addition, this procedure must process enough time in order to achieve the equilibrium state at each temperature.

For the minimization problem in the DDS model, the objective function is such as internal energy state. First, we would find the initial feasible solution and set the internal energy, and then randomly generate a new solution based on the current solution. If this new solution is satisfied all constraints and the objective function is better than the current solution, the new solution will replace the current one. Although the new solution is worse than the current one, there is still an acceptance probability, which determines to accept the worse solution. The acceptance probability is defined as $exp(- \Delta E/ t_{now})$, where ΔE is represented the difference between new solution and old solution, and t_{now} is the current temperature. We compare the acceptance probability with the random number *p*, where it is randomly generated each iteration, and its value is between 0 and 1. Therefore, the merit of the procedure can avoid the local optimum because it accepts the worse new solution with the acceptance probability.

Here, some parameters must be set to control the number of iterations at each temperature, such as α and β , where $\alpha < 1$ and $\beta > 1$. As a result, the cooling procedure and the temperature decrement execute to set $t_{now+1} = t_{now} \times \alpha$ and $b_{now+1} = b_{now} \times \beta$, where b_{now} is the number of iterations at the temperature t_{now}. The parameter t_f is the final temperature and usually set to zero, meaning at the frozen temperature and under the steady state. If the temperature reaches t_f, the approximated optimal solution will be obtained. We describe the detail procedure of Simulated Annealing method in Figure 3-4.

Then we apply the SA-based heuristic algorithm to solve the DDS model, and the core pseudo code is shown in Table 3-11. At this phase, we would generate a neighbor solution by means of moving the shares and decrypted keys of secrets to other locations. We develop our heuristic for finding new solutions due to the property of SA heuristics, which the neighbor solution is generated based on the previous solution randomly. This approach is the random-based algorithm to choose the secret first and then readjust the deployment of the shares and decrypted keys. Furthermore, we must ensure the QoS constraints met or not for each legitimated user, ex: availability and reliability. To compare each feasible solution we found, we would save the decision variables of the best ten solutions as the input of the ATSS model to evaluate the impact of the attacker. Finally, we use the SA procedure to obtain the better discrete degree of secrets solution, and compare it with the solution of the two layer mathematical model. The experiment results will be presented in chapter 4.

Figure 3-4 The Procedure of Simulated Annealing Method

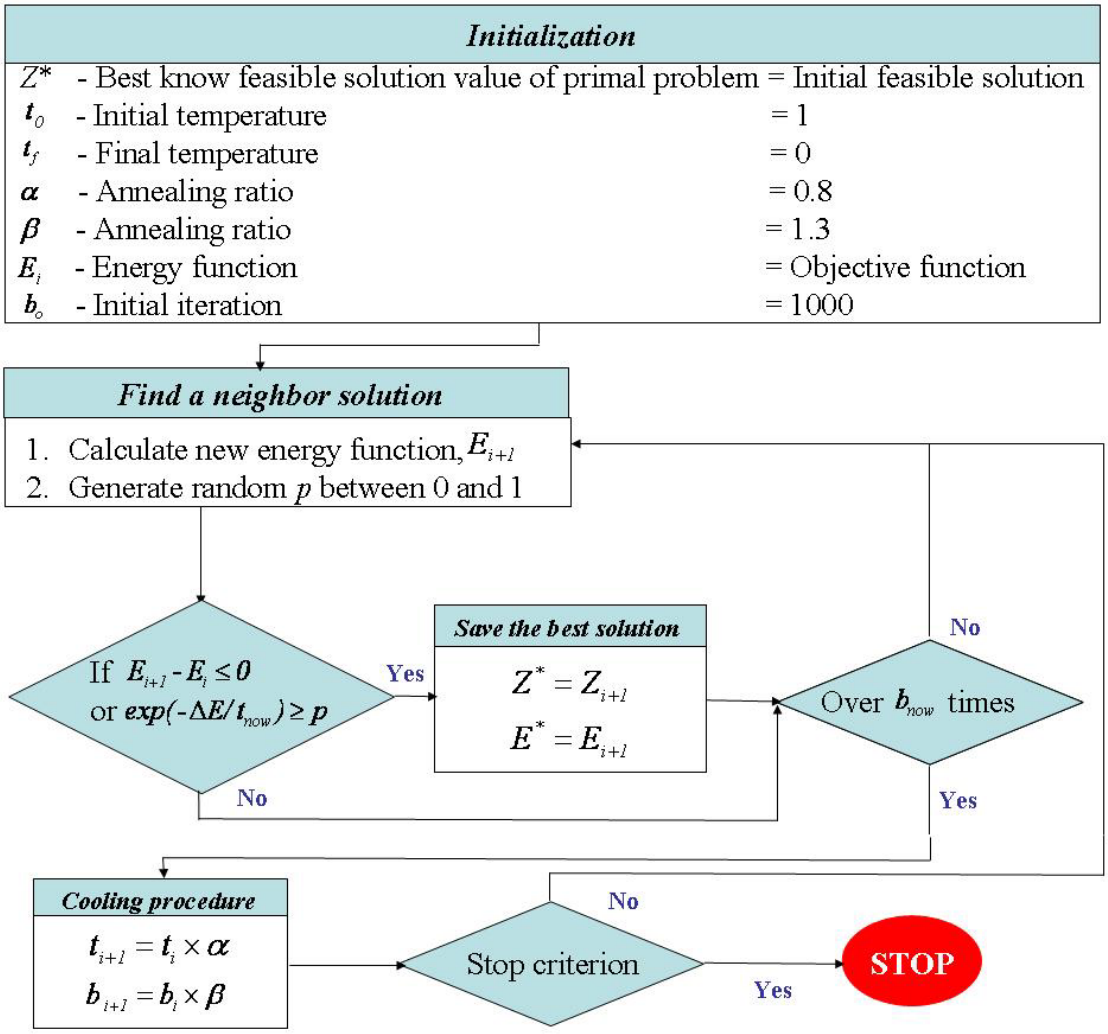

Initialization
Z^* - Best know feasible solution value of primal problem = Initial feasible solution
t_0 - Initial temperature = 1
t_f - Final temperature = 0
α - Annealing ratio = 0.8
β - Annealing ratio = 1.3
E_i - Energy function = Objective function
b_o - Initial iteration = 1000
Find a neighbor solution
1. Calculate new energy function, E_{i+1}
2. Generate random p between 0 and 1
If $E_{i+1} - E_i \le 0$ or $exp(-\Delta E/t_{now}) \ge p$
Yes
No
Save the best solution
$Z^* = Z_{i+1}$
$E^* = E_{i+1}$
Over b_{now} times
No
Yes
Cooling procedure
$t_{i+1} = t_i \times \alpha$
$b_{i+1} = b_i \times \beta$
Stop criterion
No
Yes
STOP

Table 3-11 SA-based Heuristic Algorithm for the DDS model

```
//Set the initial configuration
Set the SA parameters, $t_0, t_f, \alpha, \beta$;
//Generate the initial feasible solution
Construct the grid network topology;    // depth-in-defense
According to the delay constraints and routing constraints, choose the candidate node
which shares and keys could placed;
Calculate initial energy function $E_o$, which is calculated by Discrete Degree;
$E_o = E_{min}$; //save the initial configuration as the best solution;
$t_{now} = t_o, b_{now} = b_o$;
//Cooling procedure
While ($t_{now} > t_f$){
   While($iteration \le b_{now}$){
      Generate the random probability $p$;
      // Alter the solution configuration
      $E_{now}$= Find_ Neighborhood (); //Randomly choose Secret $\upsilon$, and redistribute
                                                  shares and decrypted keys under QoS constraints
      $\Delta E = E_{now} - E_o$;
      If( $\Delta E \le 0$  OR  $exp( - \Delta E/ t_{now}) \ge p$ ){
         $E_o = E_{now}$;
         If( $E_o \le E_{min}$ ){
            $E_{min} = E_o$;
            Record the best ten solutions;
         }
      }
      $iteration$++;
   }
   $t_{now} = t_{now} \times \alpha$;
   $b_{now} = b_{now} \times \beta$;
}
Verify the reliability constraint();
```

Chapter 4 Computational Experiments

4.1 Computational Experiments with the ATSS Model

To evaluate the performance of our proposed heuristic algorithm is effective we implement two simple algorithms for comparisons.

4.1.1 Simple Algorithm 1

The concept of the simple algorithm 1 depends on the current condition of the secret to determine that which node with the smallest weight has the highest priority to be compromised. First, we calculate the damage of the node dynamically, as the Heuristic_LR_Algorithm, and we create the Next_Attack_candidate which is the set of the neighborhood of the attack tree. The nodes in Next_Attack_candidate are the candidate targets which the attacker could choose to compromise.

In addition, we add bonus damage to those candidate nodes which might be the next target for the attacker. In such way, we consider two layer benefit rather than single layer benefit in order to obtain the potential effect. After setting the weight of the node, we apply the greedy algorithm to construct an attack tree from the attacker's initial position. If the attacker has enough budget or capability to compromise the certain node in the Next_Attack_candidate, he/she will compromise this node and update the Next_Attack_candidate then the weight is recalculated again.

The procedure would be repeated until the attack budget is exhausted. The total computational complexity of the SA_1 is $O(|N_1|^3)$. To sum up, the main idea of the simple algorithm 1 arises from the intention of the attacker to compromise nodes with the smallest weight for the most beneficial effect. The core pseudo code of the simple algorithm 1, denoted by SA_1 is shown in Table 4-1.

Table 4-1 SA$_1$ Algorithm

```
While( Attack_total_cost < ATTACK_BUDGET AND uncompromised nodes exist){
  //Calculate the weight of the node dynamically
  For each Node i which is the uncompromised node{
    For each Secret υ{
      If(Secret υ is not recovered){
        //(share_threshold[υ]<= 0 AND !decrypted_key[υ] )
        If(Node i contains the decrypted key that the attacker needs){
          $Node[i].damage+ = Secretdamage[\upsilon]^2$
        }
        //(share_threshold[υ]> 0 AND decrypted_key[υ] )
        If(Node i contains the share that the attacker needs){
          $Node[i].damage+ = \frac{Secretdamage[\upsilon]^2}{threshold[\upsilon] - basket[\upsilon]}$
        }
        //(share_threshold[υ]> 0 AND !decrypted_key[υ] )
        If(Node i contains the share that the attacker doesn't need){
          $\mathrm{Node[i].damage+} = \frac{\mathrm{Secretdamage}[\upsilon]}{2*(\mathrm{threshold}[\upsilon] - \mathrm{basket}[\upsilon])}$
        }
        //(share_threshold[υ]> 0 AND !decrypted_key[υ] )
        If(Node i contains the decrypted key that attacker doesn't need){
          $\mathrm{Node[i].damage+} = \frac{\mathrm{Secretdamage}[\upsilon]}{(\mathrm{threshold}[\upsilon] - \mathrm{basket}[\upsilon])}$
        }
      }
    }
  }
  //Calculate second layer's damage feedback to first layer
  For each node i in Next_Attack_candidate{
    For each node h which is node i's neighborhood AND still uncompromised{
      $Node[i].damage+ = Node[h].damage$ ;
    }
    $Node[i].weight = \frac{\hat{a}_i(b_i)^2}{Node[i].damage}$ ;
  }
  Choose Node i with the smallest weight node in Next_Attack_candidate
   AND the attack_cost of Node i is no more than
   (Attack_budget - Attack_total_cost);
  Compromise Node i and add to attack-tree;
  For each secret υ {
    If(Node i contains shares of secret υ )
      share_threshold[υ] - -;
    If(Node i contains decrypted key of secret υ )
      decrypted_key[υ] = true;
```

```
        }
        Attack_total_cost += Attack_cost of Node i;
        Update share_basket;
        Update decrypted_key_basket;
        Update Next_Attack_candidate;
    }
}
```

4.1.2 Simple Algorithm 2

The fundamental concept of the simple algorithm 2 is derived from the Heuristic_LR_Algorithm. In terms of the state of all secrets, we choose the sum of paths with the smallest weight which is set as the target to be recovered. As the simple algorithm 1, first, we still calculate the weight dynamically after recovering one secret and we apply Prim's algorithm to predetermine the path from the attacker's position to each node. Second, for each unrecovered secret, we sum first $k_{\upsilon}th$ the paths' weight that nodes contain shares and the certain path weight that the node contains the decrypted key, and then sort these unrecovered secrets by the weight in ascending order.

Therefore, we set the secret with the smallest weight of the sum of paths as the target, meaning that it is the most beneficial effect. Then, the uncompromised node on the chosen path must be compromised if the attack budget is sufficient. The procedure is repeated until the attack budget is exhausted or all secrets are already checked, and the computational complexity of the SA_2 is $O(|N_I| \cdot |\upsilon|^2 \cdot |m_{\upsilon}|)$. The core pseudo code of the simple algorithm 2, denoted by SA_2 is shown in Table 4-2.

Table 4-2 SA_2 Algorithm

```
While(Attack_total_cost < ATTACK_BUDGET AND unchecked secrets still exist){
        For each Node i{
            Node[i].damage = 0;
            If(Node i is compromised){
                Node[i].weight = 0;
            }
            Else{    //Only set weight to uncompromised nodes
            For each Secret $\upsilon$ {
               If(Secret $\upsilon$ is not recovered){
                  //( share_threshold[$\upsilon$]<= 0 AND !$decrypted\_key[\upsilon]$ )
                  If(Node i contains the decrypted key that the attacker needs){
                     $Node[i].damage += Secretdamage[\upsilon]^2$
                  }
                  //( share_threshold[$\upsilon$]> 0 AND $decrypted\_key[\upsilon]$ )
                  If(Node i contains the share that the attacker needs){
                     $Node[i].damage += \frac{Secretdamage[\upsilon]^2}{threshold[\upsilon] - basket[\upsilon]}$
                  }
                  //( share_threshold[$\upsilon$]> 0 AND !$decrypted\_key[\upsilon]$ )
                  If(Node i contains the share that attacker doesn't need){
                     $Node[i].damage += \frac{Secretdamage[\upsilon]}{2 \cdot (threshold[\upsilon] - basket[\upsilon])}$
                  }
                  //( share_threshold[$\upsilon$]> 0 AND !$decrypted\_key[\upsilon]$ )
                  If(Node i contains the decrypted key that attacker doesn't need){
                     $Node[i].damage += \frac{Secretdamage[\upsilon]}{2}$
                  }
               }
            }
            $Node[i].weight = \frac{\hat{a}_i(b_i)^2}{Node[i].damage}$;
         }
      }
      //Choose the secret the attacker must recover it
      $Target\_Secret = Find\_Target\_Secret();$ // shown as Table 3.5
      For each first $k_\upsilon th$ Node i contains the component of Target_Secret {
         IF($Attack\_total\_cost + path\_cost\ of\ Node\ i <= ATTACK\_BUDGET$){
            Compromise Node i and all nodes on the chosen path;
            Add these nodes to attack tree;
            $attack\_total\_cost += path\_cost\ of\ Node\ i;$
         }
      }
      For each secret $\upsilon$ {
         If(Node i contains shares of secret $\upsilon$ )
            share_threshold[$\upsilon$]--;
         If(Node i contains key of secret $\upsilon$ )
            $decrypted\_key[\upsilon] = true$;
      }
      Update share_basket;
      Update decrypt_key_basket;
}
```

4.1.3 Experiment Environment

The proposed algorithm for the ATSS Model is coded in Dev- C++ run on a PC with an Intel(R) Core(TM) 2.00GHz Duo CPU. The Iteration Counter Limit and Improve Counter Limit are set to 1000 and 50 respectively, and the initial UB is set 0. In addition, the initial step size scalar λ is set to two and halved it if objective function value still didn't improve until the counter is up to the Improve Counter Limit.

To evaluate the effect of different damage value distribution for the attacker, we design three different patterns of damage value in our system. The first is uniform distribution, which is the scope of the information value is from two to twelve, and there are the same secret numbers in each different level; the second is the normal distribution, which the damage value pattern is normally distributed, with a mean of 7 and a standard deviation of 1.6667; the third is the deterministic, whose secret damage is the same, meaning each secret is the equally important.

We design the different number of users with the certain attack budget to evaluate the performance of our system under the different QoS constrained circumstance. The more number of legitimate users exist (U1 to U5), the more reliability network operators must guarantee. Furthermore, there are three budget allocation strategies we can observe to determine which budget allocation strategy is more effective in different cases. The first strategy is uniform-based budget allocation, where each node we allocate the same defense budget. The second strategy is degree-based budget allocation, where we allocate the defense budget according to the percentage that the degree number of the node over total degree of the network. The third strategy is share-count-based budget allocation, where we allocate the defense budget depending on how many shares and decrypted keys the node contains.

In addition, we design the three different defense functions, concave, linear and convex under the B_3 allocation, and then we adopt the concave function to represent the nodal defense capability in the outer problem since the situation is close to the real environment. That is meaning that the network operators allocate too much budget on a certain nodal defense capability being useless because of marginal effect. The attacker just invests the same as the defense capability due to cost-effectiveness, otherwise, the investment is will not profitable. All parameters are shown in Table 4-3.

Table 4-3 Experiment Parameter Settings for the ATSS Model

Parameters of LR	
Parameters	Value
Iteration Counter Limit	1000
Improve Counter Limit	50
Initial Multiplier Value	$\mu_1^0, \mu_2^0, \mu_3^0, \mu_4^0, \mu_5^0 = 0$
Initial Scalar of Step Size λ	2
Platform	CPU: Intel(R) Core(TM) 2.00GHz Duo RAM: 2.5 GB
Parameters of the ATSS Model	
Parameters	Value
Number of Nodes $\lvert N_1 \rvert$	25, 64, 100
Number of Secret	$2 \cdot \lvert N_1 \rvert$
Number of User	$\left\lfloor \sqrt{\lvert N_1 \rvert} \right\rfloor, \left\lfloor \lvert N_1 \rvert / 3 \right\rfloor, \left\lfloor \lvert N_1 \rvert / 2 \right\rfloor, \left\lfloor 2\lvert N_1 \rvert / 3 \right\rfloor, \lvert N_1 \rvert$
Total Budget B	Equal to the number of nodes
Total Defense Budget A	B
Information value Distribution	Uniform Distribution (D_1) Normal Distribution (D_2) Deterministic (D_3)
Node Capacity	$1.2 \cdot \lvert \mathrm{N}_1 \rvert$
Threshold of the secret	The number of shares * 0.6
Defense Budget Distribution Strategy	Uniform-based Distribution (B_1) Degree-based Distribution (B_2) Share-count-based Distribution (B_3)
Defense Capability $\hat{a}_i(b_i)$	Concave: $2 \cdot \log(6 b_i + 1) + \varepsilon$ Linear: $2 b_i + \varepsilon$ Convex: $b_i^2 + \varepsilon$ b_i is budget allocated to node i, $\forall i \in N_1$

4.1.4 Experiment Results

We use the system vulnerability as the metric to realize the degree to which the attacker's objective is revealed, shown in Section 2.1. In order to evaluate the LR-based algorithm we proposed, we compare with SA_1 and SA_2, whereby the solution obtained from the simple algorithm 1 and the simple algorithm 2 respectively. The LR value obtained from the getting primal solution, and the LB value obtained from solving the Lagrangean relaxation process. Therefore, the optimal solution exactly exists between LR and LB so that we must calculate the gap between LR and LB by means of $\frac{\text{LB - LR}}{\text{LR}}\times 100\%$. In addition, the LR value is calculated by the percentage of the getting primal solution over total damage in our system. The improvement ratio of LR to SA_1 and SA_2 is calculated by $\frac{\text{LR-SA}_1}{\text{SA}_1}\times 100\%$ and $\frac{\text{LR-SA}_2}{\text{SA}_2}\times 100\%$. Table 4-4 to 4-7 are the value of the experiment results, and then we arrange them to Figure 4-1 to 4-11 and discuss them in the next section.

Table 4-4 Experiment Results of Small-sized Networks ($|N_I| = 25$)

Damage value	User Num	Strategy	LR(%)	Gap(%)	SA1(%)	Impro. Ratio to SA1(%)	SA2(%)	Impro. Ratio to SA2(%)
Uniform distribution	U1	B1	69.54	25.01	57.80	20.31	59.36	17.15
		B2	70.61	22.97	55.58	27.03	60.76	16.22
		B3	67.70	26.42	47.62	42.17	58.78	15.17
	U2	B1	71.51	22.30	54.68	30.78	59.93	19.32
		B2	72.70	16.84	53.28	36.44	61.00	19.18
		B3	70.20	22.41	53.45	31.34	58.78	19.42
	U3	B1	75.62	17.08	63.55	18.99	61.66	22.64
		B2	76.00	15.42	52.79	43.96	61.25	24.09
		B3	73.00	19.62	42.69	70.99	58.87	24.01

	U4	B1	77.59	14.18	65.44	18.57	61.90	25.33
		B2	78.50	10.54	66.09	18.77	64.37	21.96
		B3	74.40	19.68	49.43	50.53	60.34	23.29
	U5	B1	79.06	12.18	59.44	33.01	63.71	24.10
		B2	81.03	10.54	66.09	22.61	64.37	25.89
		B3	75.00	19.42	53.45	40.32	62.07	20.83
Normal distribution	U1	B1	64.65	28.91	39.16	65.08	50.33	28.45
		B2	65.70	29.22	47.54	38.21	47.04	39.66
		B3	63.70	26.48	40.15	58.66	55.42	14.94
	U2	B1	67.70	27.96	46.39	45.94	41.95	61.37
		B2	68.60	18.49	43.35	58.25	46.96	46.07
		B3	66.05	25.89	42.69	54.72	47.87	38.00
	U3	B1	70.60	19.54	52.71	33.94	47.87	47.50
		B2	72.20	19.26	49.26	46.97	52.13	38.87
		B3	67.70	23.95	43.68	55.00	49.75	36.07
	U4	B1	72.00	16.38	58.78	22.48	51.40	40.09
		B2	73.80	15.80	54.76	36.41	52.13	43.28
		B3	68.40	23.24	44.33	54.28	47.87	42.90
	U5	B1	73.30	14.50	51.97	41.04	54.11	35.48
		B2	74.90	13.27	58.78	29.52	51.40	48.15
		B3	70.00	23.72	48.11	45.49	49.51	41.39
Deterministic	U1	B1	66.67	22.51	56.49	18.02	57.80	15.34
		B2	68.20	15.18	53.86	26.63	57.80	17.99
		B3	65.33	20.64	51.23	27.53	58.46	11.76
	U2	B1	68.12	22.18	52.55	28.14	57.80	16.49
		B2	69.30	16.95	57.14	21.28	55.83	24.13
		B3	66.60	16.76	51.89	28.35	56.49	17.91
	U3	B1	70.67	17.41	57.80	22.26	61.08	15.69
		B2	72.67	14.73	63.05	15.24	59.11	22.93
		B3	68.67	16.67	51.89	32.34	57.80	18.80
	U4	B1	72.67	14.77	63.05	15.24	63.71	14.06
		B2	74.67	11.75	65.68	13.68	65.68	13.68
		B3	69.00	16.51	54.52	26.57	57.80	19.38
	U5	B1	74.67	12.21	65.68	13.68	65.68	13.68
		B2	75.30	12.19	67.00	12.40	64.37	16.98
		B3	70.00	15.53	54.52	28.40	57.14	22.50

Table 4-5 Experiment Results of Medium-sized Networks ($\left|N_I\right| = 64$)

Damage value	User Num	Budget	LR(%)	Gap(%)	SA1(%)	Impro. Ratio to SA1(%)	SA2(%)	Impro. Ratio to SA2(%)
Uniform distribution	U1	B1	73.25	22.62	62.49	17.23	56.27	30.18
		B2	75.14	18.65	67.37	11.53	62.04	21.11
		B3	67.59	29.54	51.94	30.13	54.94	23.03
	U2	B1	74.36	21.44	62.49	19.01	57.38	29.59
		B2	75.14	20.88	63.49	18.36	62.71	19.82
		B3	69.00	29.47	56.83	21.42	50.28	37.24
	U3	B1	74.36	21.63	59.60	24.77	54.27	37.01
		B2	75.80	19.50	64.04	18.37	58.49	29.60
		B3	71.48	23.84	57.27	24.81	53.50	33.61
	U4	B1	75.25	17.46	59.60	26.26	54.27	38.65
		B2	77.40	14.67	62.60	23.65	63.60	21.71
		B3	72.90	22.63	57.27	27.29	51.61	41.25
	U5	B1	79.36	14.95	65.48	21.19	65.04	22.01
		B2	81.02	13.36	66.26	22.28	62.71	29.20
		B3	76.00	16.04	61.04	24.50	53.94	40.90
Normal distribution	U1	B1	69.12	25.77	49.75	38.92	58.21	18.74
		B2	69.73	24.66	52.21	33.57	53.92	29.32
		B3	66.91	29.28	50.61	32.20	50.37	32.85
	U2	B1	70.96	23.71	48.65	45.84	57.48	23.45
		B2	71.69	22.36	48.65	47.36	46.69	53.54
		B3	67.50	29.76	44.12	53.00	60.29	11.95
	U3	B1	70.90	22.67	52.21	35.81	56.86	24.69
		B2	72.30	21.50	55.88	29.39	58.21	24.21
		B3	67.70	28.24	52.08	29.98	56.00	20.88
	U4	B1	71.30	22.75	48.65	46.55	57.48	24.05
		B2	73.00	18.97	57.48	27.01	57.72	26.47
		B3	68.20	29.09	44.12	54.59	55.64	22.58
	U5	B1	75.12	18.50	56.37	33.26	47.67	57.58
		B2	75.98	17.24	59.56	27.57	61.15	24.25
		B3	70.20	26.49	54.66	28.44	46.45	51.14
	U1	B1	70.37	17.78	55.56	26.67	59.26	18.75
		B2	71.70	14.85	58.33	22.91	55.56	29.06
		B3	64.81	24.78	47.22	37.25	51.85	25.00

Deterministic	U2	B1	72.22	18.20	62.96	14.71	60.19	20.00
		B2	73.30	13.99	67.59	8.44	58.33	25.66
		B3	66.40	23.79	48.15	37.91	52.78	25.81
	U3	B1	73.15	15.80	58.33	25.40	62.04	17.91
		B2	74.20	16.15	59.26	25.21	53.70	38.17
		B3	68.52	19.42	53.70	27.59	51.85	32.14
	U4	B1	73.15	18.60	63.89	14.49	57.41	27.42
		B2	74.20	16.41	59.26	25.21	53.70	38.17
		B3	69.20	20.31	58.33	18.63	53.70	28.86
	U5	B1	74.07	16.13	60.19	23.08	64.81	14.29
		B2	76.00	11.25	71.30	6.60	59.26	28.25
		B3	71.30	16.62	54.63	30.51	50.93	40.00

Table 4-6 Experiment Results of Large-sized Networks ($|N_I|$ = 100)

Damage value	User Num	Budget	LR(%)	Gap(%)	SA1(%)	Impro. Ratio to SA1(%)	SA2(%)	Impro. Ratio to SA2(%)
Uniform distribution	U1	B1	74.83	28.59	67.41	11.02	64.03	16.88
		B2	77.31	25.53	74.75	3.42	63.45	21.85
		B3	69.72	38.19	66.01	5.63	49.01	42.26
	U2	B1	78.71	22.49	65.92	19.40	71.70	9.78
		B2	80.78	20.74	72.03	12.14	63.61	26.98
		B3	72.69	32.54	58.99	23.22	53.47	35.96
	U3	B1	79.37	22.74	70.79	12.12	68.81	15.35
		B2	81.68	18.73	73.02	11.86	70.13	16.47
		B3	74.75	29.50	58.17	28.51	53.80	38.96
	U4	B1	80.61	21.52	70.79	13.87	68.81	17.15
		B2	82.26	18.52	77.56	6.06	63.70	29.15
		B3	77.56	25.80	63.78	21.60	55.78	39.05
	U5	B1	82.76	17.66	76.73	7.85	71.37	15.95
		B2	83.09	17.43	74.83	11.03	68.98	20.45
		B3	78.14	24.66	63.28	23.47	44.47	75.70
	U1	B1	71.92	31.21	61.41	17.13	55.96	28.52
		B2	73.00	29.71	61.50	18.69	59.81	22.05
		B3	68.26	36.38	62.07	9.98	49.11	39.01

Normal distribution	U2	B1	76.00	23.97	68.83	10.42	55.87	36.03
		B2	77.40	23.58	68.54	12.92	55.12	40.43
		B3	72.68	28.98	51.92	39.96	46.48	56.36
	U3	B1	76.62	23.90	68.83	11.32	55.87	37.14
		B2	78.00	23.80	68.45	13.95	64.79	20.39
		B3	73.71	25.27	63.29	16.47	47.89	53.92
	U4	B1	78.12	19.94	69.58	12.28	61.69	26.64
		B2	79.90	24.08	74.27	7.58	66.29	20.53
		B3	74.46	29.87	65.63	13.45	47.89	55.49
	U5	B1	78.97	18.63	71.46	10.51	63.10	25.15
		B2	81.50	21.33	67.98	19.89	56.24	44.91
		B3	74.55	27.24	71.08	4.89	51.83	43.84
Deterministic	U1	B1	70.07	28.43	64.63	8.42	55.10	27.16
		B2	70.75	35.59	66.67	6.12	55.78	26.83
		B3	63.95	42.86	48.30	32.39	47.62	34.29
	U2	B1	72.11	27.45	61.90	16.48	65.31	10.42
		B2	73.50	24.95	72.79	0.98	57.82	27.11
		B3	64.63	36.32	52.38	23.38	53.06	21.79
	U3	B1	73.47	24.05	66.67	10.20	62.59	17.39
		B2	74.83	25.58	72.79	2.80	56.46	32.53
		B3	65.99	35.34	56.46	16.87	53.74	22.78
	U4	B1	75.51	25.31	65.99	14.43	65.31	15.63
		B2	77.00	24.19	64.63	19.15	62.59	23.03
		B3	68.03	32.33	53.06	28.21	54.42	25.00
	U5	B1	76.87	21.76	63.27	21.51	66.67	15.31
		B2	78.00	26.51	69.39	12.41	68.03	14.66
		B3	70.07	33.92	54.42	28.75	42.86	63.49

Table 4-7 Experiment Results under Different Defense Capability Function (B3)

Defense Function (Under B3)	Damage value	User Num	LR(%)	Gap(%)	SA1(%)	Impro. Ratio to SA1(%)	SA2(%)	Impro. Ratio to SA2(%)
Concave Function	Uniform distribution	U1	70.43	31.36	51.85	35.83	52.74	33.54
		U2	73.32	26.85	56.03	30.85	54.74	33.94
		U3	75.94	23.61	58.62	29.55	54.92	38.27
		U4	76.05	23.59	60.44	25.84	56.40	34.84
		U5	76.87	23.11	61.58	24.82	53.74	43.04
	Normal distribution	U1	68.26	32.93	54.06	26.26	53.57	27.42
		U2	71.48	24.73	57.12	25.14	52.75	35.50
		U3	73.36	29.80	58.55	25.30	52.39	40.03
		U4	74.34	25.40	60.47	22.94	51.94	43.13
		U5	75.23	24.35	65.16	15.47	53.94	39.49
	Deterministic	U1	67.27	30.38	45.45	48.00	50.61	32.93
		U2	68.18	31.78	44.24	54.11	48.79	39.75
		U3	70.61	28.20	47.88	47.47	51.21	37.87
		U4	71.52	27.09	46.97	52.26	50.91	40.48
		U5	71.82	28.14	54.24	32.40	48.48	48.13
Linear Function	Uniform distribution	U1	67.85	31.36	48.08	41.13	53.92	25.83
		U2	71.24	26.85	51.22	39.09	58.51	21.76
		U3	75.13	23.61	55.74	34.80	60.03	25.16
		U4	79.31	23.59	59.92	32.37	60.36	31.39
		U5	80.64	23.11	66.54	21.19	64.21	25.59
	Normal distribution	U1	66.62	32.93	49.45	34.72	56.51	17.89
		U2	70.54	24.73	56.87	24.03	54.35	29.80
		U3	73.85	29.80	57.61	28.19	56.79	30.03
		U4	75.85	25.40	59.93	26.55	60.71	24.93
		U5	78.62	24.35	61.57	27.70	64.30	22.27
	Deterministic	U1	62.12	30.38	40.61	52.99	52.12	19.19
		U2	67.58	31.78	46.36	45.75	54.85	23.20
		U3	70.00	28.20	45.76	52.98	57.88	20.94
		U4	72.42	27.09	51.21	41.42	61.82	17.16
		U5	76.36	28.14	56.36	35.48	63.64	20.00
	Uniform distribution	U1	52.31	32.93	28.56	83.14	40.15	30.28
		U2	57.45	24.73	33.17	73.19	43.00	33.59
		U3	61.73	29.80	38.27	61.30	44.59	38.43

Convex Function		U4	63.85	25.40	41.62	53.43	49.53	28.91
		U5	69.48	24.35	47.33	46.81	52.92	31.30
	Normal distribution	U1	50.70	31.36	30.57	65.86	35.05	44.67
		U2	53.26	26.85	36.38	46.39	38.71	37.57
		U3	58.14	23.61	42.38	37.21	43.04	35.08
		U4	61.40	23.59	46.74	31.35	44.89	36.77
		U5	65.62	23.11	51.37	27.74	50.67	29.51
	Deterministic	U1	44.85	30.38	30.00	49.49	36.36	23.33
		U2	50.61	31.78	29.39	72.16	40.91	23.70
		U3	54.55	28.20	36.97	47.54	41.82	30.43
		U4	56.67	27.09	41.52	36.50	42.12	34.53
		U5	61.52	28.14	47.58	29.30	50.91	20.83

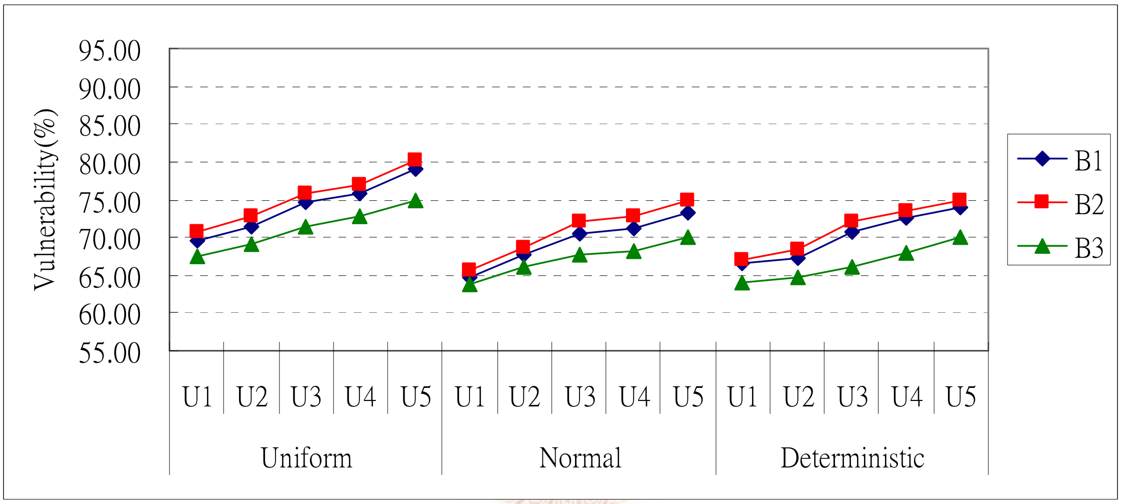


Figure 4-1 The Network Vulnerability under Different Numbers of Users ($|N_I|$ = 25)

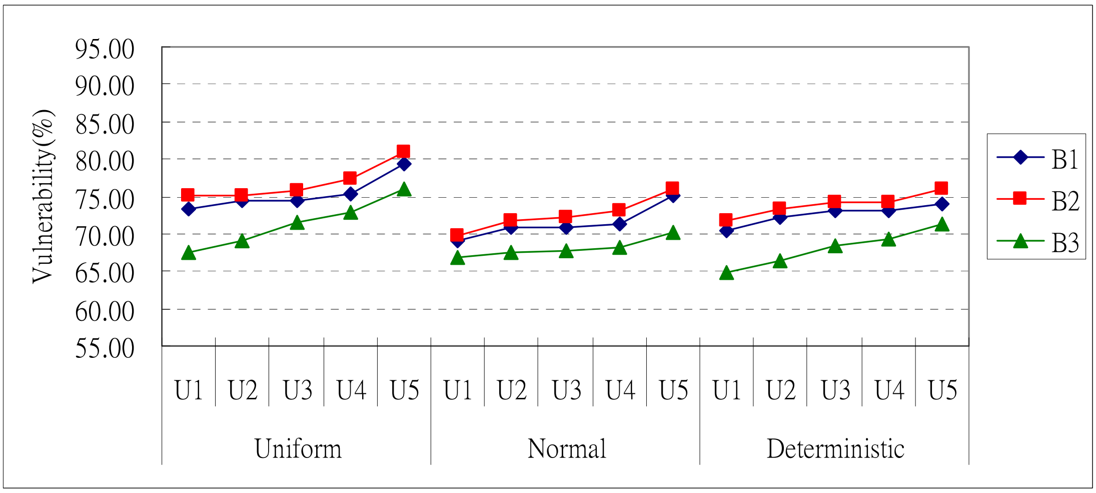


Figure 4-2 The Network Vulnerability under Different Numbers of Users ($|N_I|$ = 64)

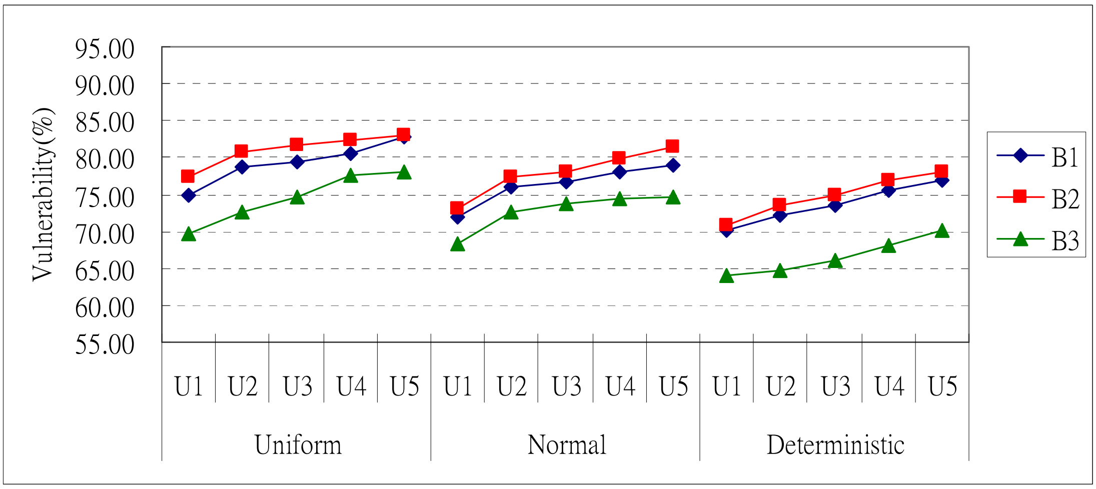


Figure 4-3 The Network Vulnerability under Different Numbers of Users ($|N_I|$ = 100)

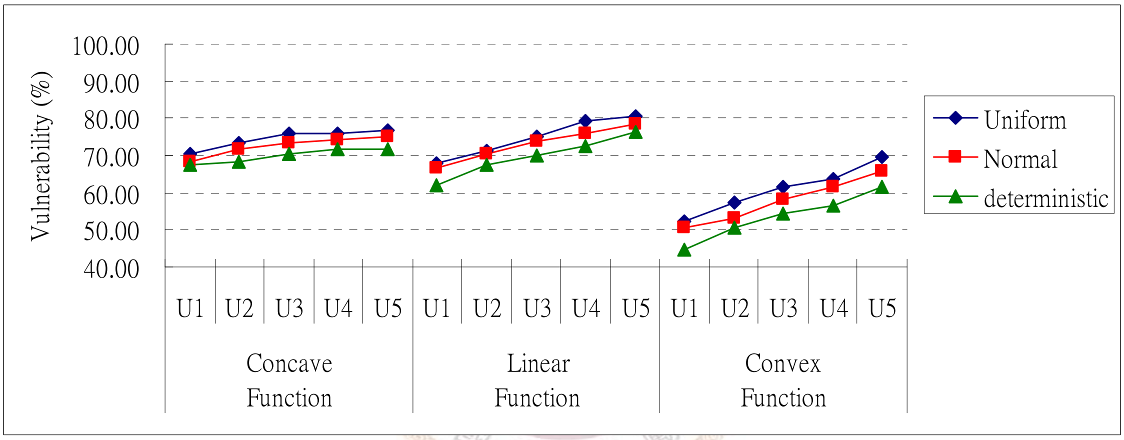


Figure 4-4 The Network Vulnerability under the Different Defense Function (B3)

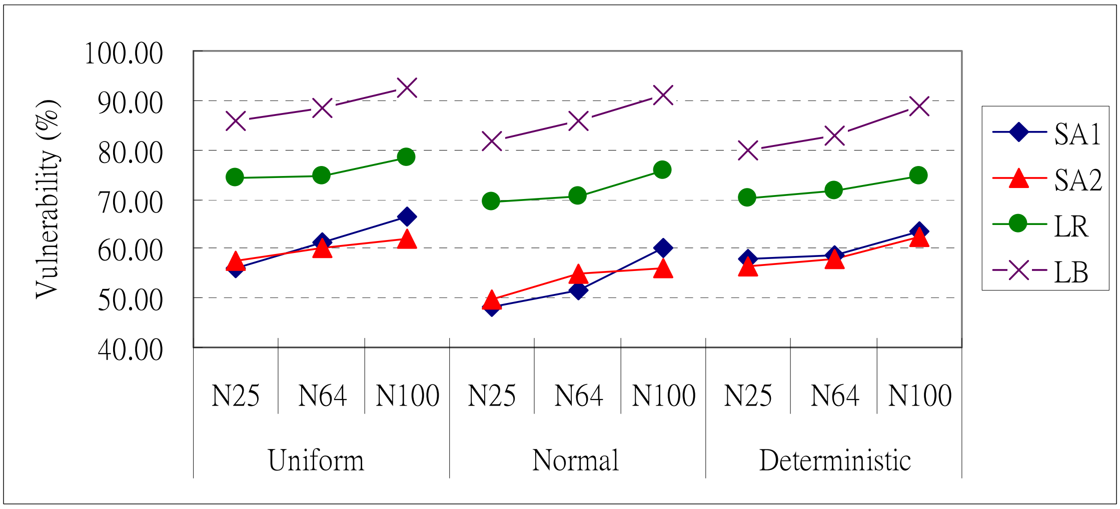


Figure 4-5 Vulnerability of Different Network Sizes and Damage Distributions

4.1.5 Discussion of Results

Figure 4-1 to Figure 4-3 indicates the vulnerability of the targeted network under different numbers of nodes, numbers of users, damage value patterns with the certain attack budget equaled the defense budget.

From these figures, we observe that the target network with budget allocation strategy B_3 is the lowest vulnerability and the most robust in all cases. It performs more outstanding than other strategies since the defense resource is allocated according to the importance of each node. The more shares and keys nodes contain, the more probability are chosen as targets by attackers. Attackers must consume their budget on these high defense capability nodes so that they just enable to recover an amount of secrets. Number of compromised nodes would decrease due to the B_3 strategy.

Furthermore, the network vulnerability of the B_1 strategy outperforms the B_2 strategy because of the wrong defense resource allocation under the B_2 strategy. The probable reason is that the node with high degree might probably contain few of shares or relatively valueless shares for the attacker, so the defense resource on this node becomes useless. The consequence results in more secrets recovered with the same attack budget by the attacker easily. Although the condition of wrong defense allocation still occurs under the B_1 strategy, the uniform-based allocation can reduce the impact of wrong defense allocation.

With the growth of the network, the difference between the B_1, B_2 and B_3 increase significantly for each information value distribution. Because of the larger the network size, the more targets the attacker could choose. In order to reveal maximal system damage, the attacker will consider the ratio of the attack cost to the profit gained to avoid the node with too high defense capability. For this reason, the influence of wrong defense allocation would be magnified for network vulnerability with the growth of the network size.

The trend of network vulnerability would rise if the system must provide more users QoS requirements. Due to the higher system reliability being achieved, network operators must transfer some budget from defense budget. In addition, the number of users might affect defense-in-depth of the network because operators must construct more links to achieve reliability for O-D pairs. Generally, it is much favorable for the attacker to cause system damage since the part of defense resource allocated to decrease the random error of the link.

Theoretically, the different information value distributions also affect the network vulnerability. For example, the D1 distribution under different scenarios is usually the highest vulnerability. The phenomenon indicates there are more quantities of high information value in the targeted network, thus, the attacker would trend to recover those types of secrets. For different information value distributions, the decision the attacker adopts might be dissimilar. Under the scenarios of the D3, because all information value is the same, meaning equally important, the attacker chooses the targets according to what kind of shares and decrypted keys he already obtains.

Figure 4-4 shows the effect of different defense capability function under different damage value distribution scenarios. Regardless of the kind of defense capability functions, the rank of system vulnerability is Uniform distribution, Normal distribution and Deterministic, respectively. Under different QoS constraints, the variation of vulnerability with concave function is the lowest, with convex function being the highest, and with linear function being between them. That is network operators invest enough defense resource to enhance capability effectively by convex and linear form, however, the more resource become useless by concave form because of the marginal effect.

Figure 4-5 illustrates the performance of our proposed LR-based algorithm, simple algorithm 1, simple algorithm 2, and the gap between LRs and LBs. Each point is calculated by the average network vulnerability of different numbers of users and different defense allocation strategies under the same network size and information value distribution. We could observe the network vulnerability of our proposed algorithm is always higher than simple algorithm 1 and simple algorithm 2 among damage value distributions and the average improvement ratio to them are about 23.58 % and 27.13 %, respectively. Although the performance of SA1 and SA2 are uncertain, the time complexity of SA1 is simpler than SA2. Therefore, the SA1 is the better choice if the consideration of the complexity is considered. In addition, we could observe the performance of them is approximately identical under the deterministic damage value. The possible reason is they would choose the same targeted node to compromise since they have no idea to decide which secret to be recovered first.

There is the optimal solution between LRs and LBs, and the average gap is no more than 20.17% in our proposed algorithm. Furthermore, the network vulnerability increases with the growth of network size since the more choices for attackers to achieve their objective easily are.

4.2 Computational Experiments with the NPDS Model

4.2.1 Experiment Environment

The proposed algorithm for the NPDS Model is coded in Dev- C++ run on a PC with an Intel(R) Core(TM) 2.00GHz Duo CPU. The Iteration Counter Limit, Improve_Secret_Counter and Improve_Topology_Counter are set to 160, 8 and 8 respectively. The initial step size coefficient, θ and λ , are set to 0.5 and 0.25, and they are halved if the network vulnerability doesn’t improve after being over Improve_Topology_Counter or Improve Counter Limit.

According to the result of the ATSS model, we observe the share-count-based strategy B_3 is the best defense budget allocation of the three given strategies. Hence, we design the different number of nodes with the strategy B_3 and the number of users (U2) under different damage value distributions. We adopt the concave defense capability function to be close to real world because marginal effect decreased with the addition of defense budget.

The network operator would not only adjust the condition of recovered secrets by putting shares and decrypted key on more secure locations, but also try to strength the depth of the network and reallocate the defense budget on each node after the behavior of attacking. After each attack procedure, the network operator applies these adjustment mechanisms for the NPDS model.

Here, there are three reallocation strategies we adopted, denoted as uniform-based, degree-based (attack tree), and damage-based redistributions. We observe the network environment and extract some defense budget from uncompromised nodes and reallocate to compromised nodes with reallocation strategies. Furthermore, we develop the other compared heuristic algorithm, denoted Defense_Level_Adjustment_Only (DLAO), in order to evaluate the performance of our proposed algorithm.

The other unmentioned parameters are shown in Table 4-8 in detail.

Table 4-8 Experiment Parameter Setting for the NPDS Model

Parameters of Adjustment Procedure	
Parameters	Value
Iteration Counter Limit	160
Improve Counter Limit	8
Improve_Topology_Counter	8
Initial Scalar of Step Size θ	0.5
Initial Scalar of Step Size λ	0.25
Platform	CPU: Intel(R) Core(TM) 2.00GHz Duo RAM: 2.5 GB OS: Microsoft Windows XP SP3
Parameters of the NPDS Model	
Parameters	Value
Number of Nodes $\lvert\mathrm{N}_1\rvert$	25, 64, 100
Number of Secret	$2\cdot\lvert N_l\rvert$
Number of User	$\left\lfloor \lvert N_l\rvert / 3 \right\rfloor$
Node Capacity	$1.2\cdot\lvert N_l\rvert$
Maximum Allowable End-to-End Delay	2 (sec)
The disjoint path requirement	1to2
The material type of θ_l	$\theta_l \in \{1,2,...,\vartheta\}$
The tolerate risk of the random error	0.9
Total Budget B	Equal to the number of nodes
Total Defense Budget A	Equal to the total budget B
Information value Distribution	Uniform Distribution (D_1) Normal Distribution (D_2) Deterministic (D_3)
Initial Budget Allocation Strategy	Share-count-based Distribution (B_3)
Reallocation Strategy	Uniform-based reallocation (B_1) Degree-based reallocation (B_2) Damage-based reallocation (B_3)
Defense Capability $\hat{a}_i(b_i)$	$2\cdot\log(6b_i+1)+\varepsilon$, b_i is the budget allocated to node *i*, $\forall i \in N_l$

4.2.2 Experiment Results

The initial network vulnerability is obtained from the share-count based defense allocation strategy under the QoS (U2) requirements because the B3 defense allocation strategy is the most robust among all defense strategies. In this experiment, we use the same metric as the ATSS model to evaluate the performance of the targeted network. The NPDS Vulnerability is the improvement of initial system vulnerability with our proposed algorithm, and DLAO Vulnerability is the improvement of initial network vulnerability with only defense resource reallocation heuristic algorithm.

In this experiment, we further adopt three reallocation strategies to adjust the nodal defense capability, as ATSS model defense allocation strategies, and evaluate the performance of these strategies with our proposed algorithm and the DLAO heuristic algorithm. The improvement ratio is calculated by $\frac{\text{NPDS Vuln.- Init. Vuln.}}{\text{Init. Vuln.}} \times 100\%$ and $\frac{\text{DLAO Vuln.- Init. Vuln.}}{\text{Init. Vuln.}} \times 100\%$, respectively. The experiment results are shown in Table 4-9 in detail.

Table 4-9 The Experiment Results for the NPDS Model

Damage value	Number of Users	Reallocation Strategy	Init. Vuln. (%)	NPDS. Vuln.(%)	Imp. Ratio of NPDS Vuln. (%)	DLAO (%)	Imp. Ratio of DLAO Vuln. (%)
Uniform distribution	Node 25	B1	71.25	57.40	19.44	63.15	11.36
		B2		55.66	21.88	63.61	10.73
		B3		57.67	19.06	64.18	9.92
	Node 64	B1	71.94	54.73	23.91	66.29	7.84
		B2		60.27	16.22	66.75	7.22
		B3		62.56	13.04	67.23	6.55

	Node 100	B1	73.17	53.30	27.15	69.50	5.01
		B2		53.80	26.47	70.21	4.04
		B3		58.71	19.75	69.56	4.92
Normal distribution	Node 25	B1	64.89	50.20	22.64	59.16	8.82
		B2		51.74	20.27	59.75	7.91
		B3		53.26	17.92	61.14	5.77
	Node 64	B1	67.61	50.36	25.50	61.85	8.52
		B2		51.14	24.36	62.61	7.39
		B3		52.43	22.45	62.73	7.22
	Node 100	B1	69.58	50.21	27.84	65.11	6.43
		B2		51.44	26.08	66.74	4.08
		B3		55.50	20.23	68.23	1.95
Deterministic	Node 25	B1	62.98	46.82	25.66	58.45	7.19
		B2		49.20	21.88	58.30	7.42
		B3		51.01	19.00	59.18	6.03
	Node 64	B1	64.04	46.37	27.59	59.70	6.79
		B2		49.84	22.17	61.58	3.85
		B3		51.34	19.83	61.41	4.10
	Node 100	B1	68.98	49.80	27.80	64.61	6.33
		B2		52.75	23.53	65.66	4.81
		B3		53.60	22.29	67.29	2.45

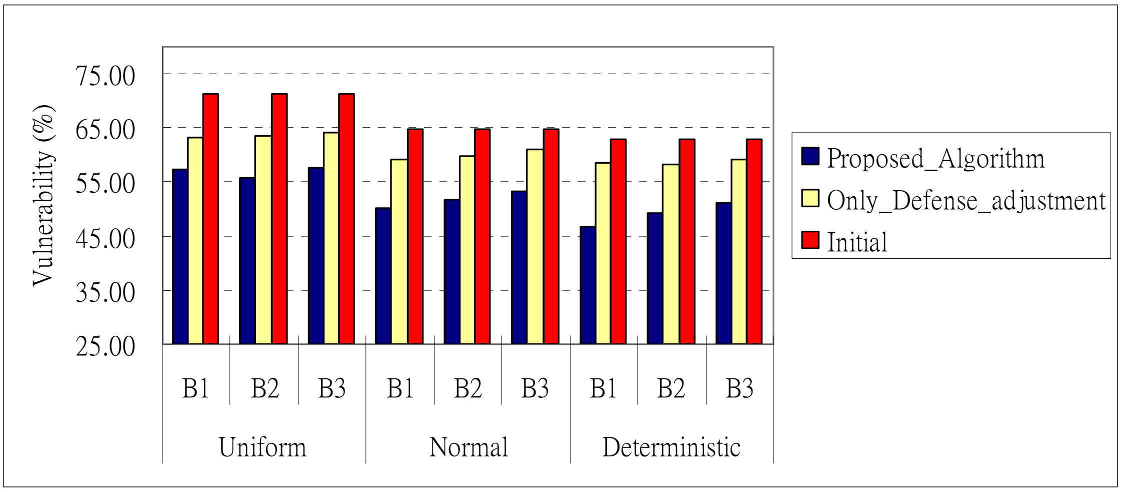


Figure 4-6 The Improvement under Different Reallocation Strategies ($|N_I| = 25$)

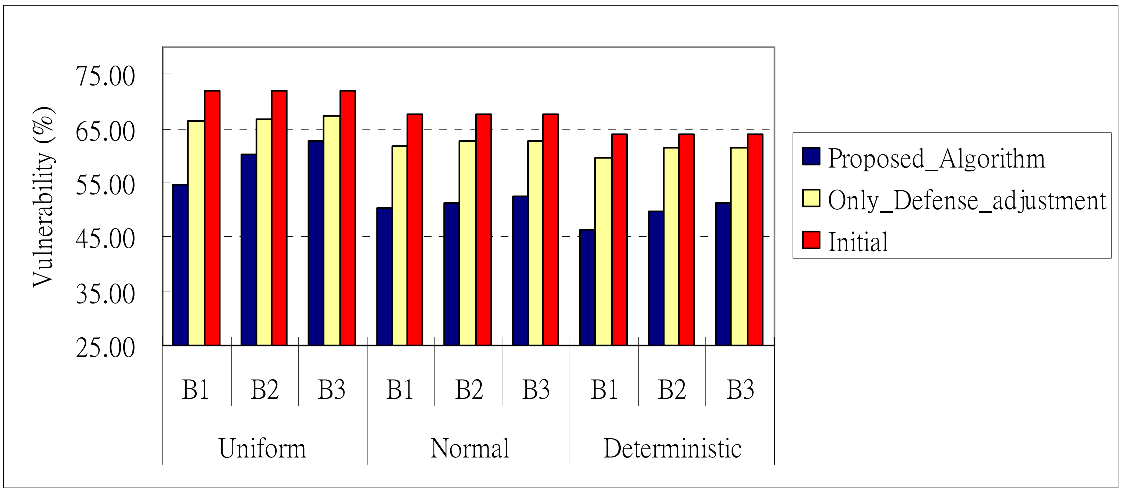


Figure 4-7 The Improvement under Different Reallocation Strategies ($|N_l|$ = 64)

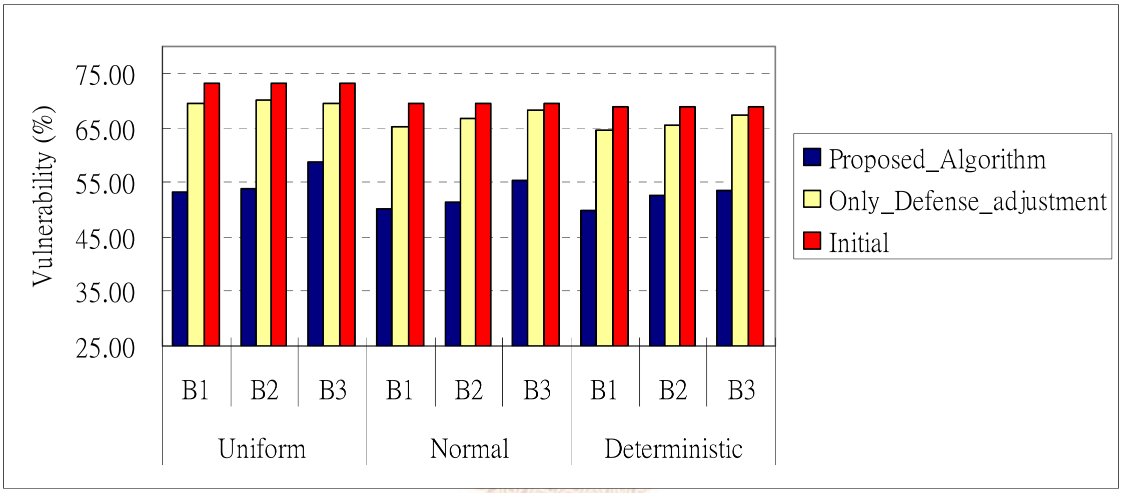


Figure 4-8 The Improvement under Different Reallocation Strategies ($|N_l|$ = 100)

4.2.3 Discussion of Results

Figure 4-6 to Figure 4-8 show the vulnerability of the targeted network under different numbers of nodes, different damage value patterns under the same QoS requirements. From these figures, we could make some observations that our proposed algorithm outperforms Defense_Level_Adjustment_Only in all cases. In addition, the difference of them increases significantly with the growth of the network. The possible reason is the more the network size is, the more the amount of choices to attack is leading to the limited improvement of DLAO heuristic algorithm. Therefore, the effective strategy against the vulnerability is enhancing defense-in-depth of the network and distributing shares over the secure location simultaneously rather than only adjusting defense resource.

To discuss the reallocation strategy further, we observe the performance of reallocation B3 strategy is lower improvement than other reallocation strategies and is contrary to the initial defense allocation strategy. The result indicates the marginal defense capability decrease with the addition of defense budget because compromised nodes which almost contain quantities of shares are already allocated more resource under initial budget allocation strategy B3.

According to this finding, the guideline of the reallocation strategy is to allocate resource on overall important nodes to make them enough defense capability rather than extremely reinforcing the certain nodal defense capability.

The average improvement ratios to initial network vulnerability for our proposed algorithm and Defense_Level_Adjustment_Only are 22.37 % and 6.47 %, respectively. To sum up, we can induce the several mechanisms is truly better than single mechanism.

4.3 Computational Experiments with the DDS Model

In this section, we adopt the best ten solutions of the DDS model as the input of the ATSS model. As a result, we can obtain the improvement between initial vulnerability and new vulnerability of the best ten solutions, and then we use these decision variables as the second solution approach. The procedure is shown in Figure 4-9. Moreover, the result of the NPDS model would be compared with the DDS model under the same scenarios, or we also combined the DDS model with the NPDS model to obtain the better defense strategy.

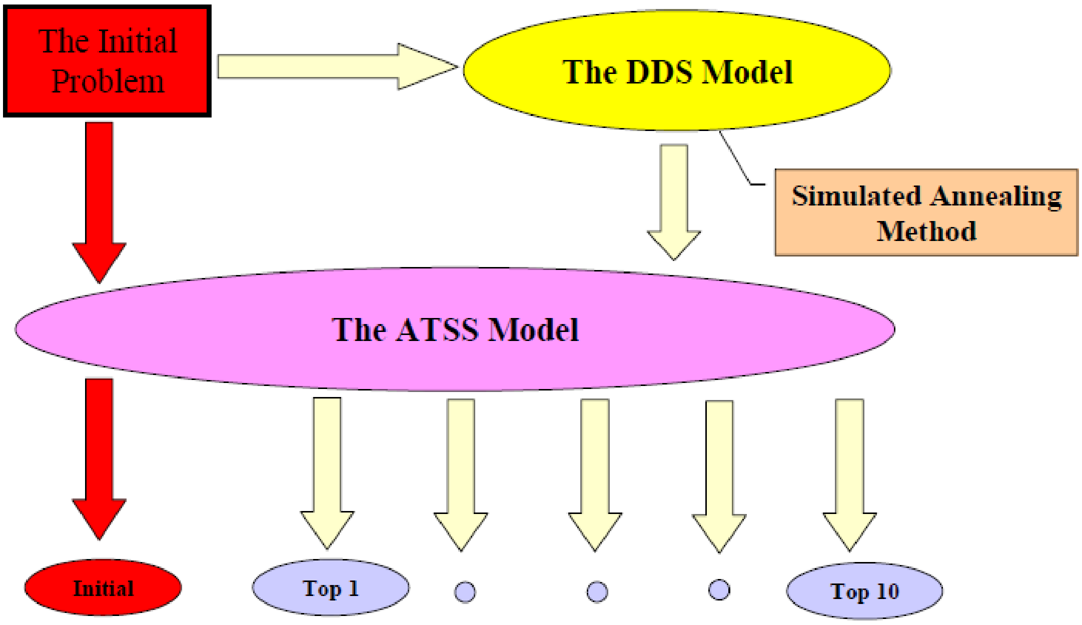


Figure 4-9 The procedure of the second solution approach

4.3.1 Experiment Environment

Here, we set the SA parameter α to 0.8, β to 1.3, the initial temperature T_0 to 1 and the frozen temperature T_f to 0. The initial iteration counter b_0 is set to 1000, and the SA procedure executes b_0 times at each temperature. After setting these parameters, we process the SA procedure and save the best ten "Discrete Degree" solutions then put them into the ATSS model. Thus, we use the concave defense capability function under the different damage value distribution to evaluate the results of the DDS model. As same as previous experiments, the three defense resource allocation strategies are used, uniform-based, degree-based and share-count-based. The remaining unmentioned parameters are shown in Table 4-10 in detail.

Table 4-10 Experiment Parameter Setting for the DDS Model

Parameters of SA Procedure	
Parameters	Value
Initial Temperature	1
Initial Iteration	1000

Final Temperature	0
Cooling Parameter	$\alpha = 0.8$ $\beta = 1.3$

Parameters of the DDS Model

Parameters	Value
Number of Nodes $\lvert \mathrm{N}_1 \rvert$	25, 64, 100
Number of Secret	$2 \cdot \lvert N_l \rvert$
Number of User	$\left\lfloor {\lvert N_l \rvert} / {3} \right\rfloor$
Node Capacity	$1.2 \cdot \lvert N_l \rvert$
Maximum Allowable End-to-End Delay	2 (sec)
The disjoint path requirement	1to2
The material type of θ_l	$\theta_l \in \{1,2,...,\vartheta\}$
The tolerate risk of the random error	0.9
Total Budget B	Equal to the number of nodes
Information value Distribution	Uniform Distribution (D_1) Normal Distribution (D_2) Deterministic (D_3)
Defense Allocation Strategy	Uniform-based Reallocation (B_1) Degree-based Reallocation (B_2) Damage-based Reallocation (B_3)
Defense Capability $\hat{a}_i(b_i)$	$2 \cdot \log(6b_i + 1) + \varepsilon$, b_i is the budget allocated to node i, $\forall i \in N_l$

4.3.2 Experiment Results

We propose the Simulated Annealing method to solve the DDS model and exploit the best ten promising solutions of the decision variables as the input of the ATSS model. Here, there are three defense allocation strategies to evaluate the performance of the attacker under different damage value distribution. The improvement ratio of the DDS model is calculated by $\frac{\text{DDS Vuln.- Init. Vuln.}}{\text{Init. Vuln.}} \times 100\%$. The Init. Vuln. value represents the network vulnerability under initial defense budget allocation strategies, and the DDS. Vuln. value is the improved vulnerability from the best ten solution strategies. In addition, we compare the result of DDS model with previous two layer mathematical model.

Table 4-11 Experiment Results for the DDS model

Network Size	Damage value	Defense Strategy	Init. Vuln. (%)	Top1 Vuln. (%)	Top2 Vuln. (%)	Top3 Vuln. (%)	Top4 Vuln. (%)	Top5 Vuln. (%)	Top6 Vuln. (%)	Top7 Vuln. (%)	Top8 Vuln. (%)	Top9 Vuln. (%)	Top10 Vuln. (%)
Node 25	Uniform	B1	75.85	59.59	59.92	59.92	60.08	59.43	62.38	60.58	61.23	62.71	65.01
		B2	76.00	63.86	60.41	66.32	63.70	65.67	63.20	67.14	65.01	67.64	70.92
		B3	72.69	64.68	68.79	64.35	66.98	65.67	66.65	66.65	66.49	67.31	69.44
	Normal	B1	70.66	55.09	54.21	58.95	57.19	62.98	64.39	62.46	62.46	63.16	62.81
		B2	71.67	59.30	61.75	67.89	67.37	64.91	66.32	70.00	69.16	71.58	70.88
		B3	70.04	55.26	60.18	63.86	66.67	70.03	68.42	66.84	69.47	68.42	70.03
	Deterministic	B1	72.10	52.59	53.93	52.59	53.93	52.59	53.93	53.93	56.59	56.59	56.59
		B2	72.15	53.93	57.93	60.59	55.26	61.93	60.59	63.26	59.26	60.59	61.93
		B3	70.34	59.26	57.93	56.59	63.26	59.26	60.59	61.93	63.26	65.93	64.59
Node 64	Uniform	B1	78.55	60.00	61.07	62.54	63.11	65.38	66.52	63.79	63.22	63.79	65.04
		B2	79.34	59.70	61.41	58.80	60.50	61.07	59.82	64.93	63.56	63.68	65.95
		B3	74.46	60.16	60.73	63.11	61.63	60.50	61.63	60.84	63.45	64.25	64.25
	Normal	B1	73.72	56.13	57.72	59.31	59.44	59.44	63.97	63.24	62.99	61.64	63.24
		B2	75.86	56.86	58.95	58.82	57.23	60.66	59.07	60.54	61.40	62.99	63.24
		B3	72.00	58.95	60.29	58.95	62.38	60.54	60.78	60.54	62.13	62.99	62.62
	Deterministic	B1	72.19	54.06	55.91	54.06	54.98	56.83	56.83	56.83	55.91	54.98	57.76
		B2	73.50	52.78	54.63	53.70	55.56	56.48	55.56	56.48	57.41	59.26	58.33
		B3	70.98	54.63	53.70	57.41	55.56	54.63	55.56	55.56	57.41	57.41	59.26
Node 100	Uniform	B1	78.71	54.02	57.00	57.24	56.42	58.98	58.73	61.37	58.32	58.73	63.43
		B2	79.78	56.83	55.18	60.54	62.19	61.04	64.59	60.54	66.15	63.68	63.02
		B3	75.22	59.97	61.04	60.46	59.80	61.12	62.52	65.25	63.10	66.07	71.76
	Normal	B1	76.00	54.51	53.01	54.60	56.29	55.35	56.01	57.42	57.61	61.55	60.24
		B2	77.40	53.29	55.35	54.79	57.61	58.36	56.20	58.08	60.61	64.65	66.72
		B3	72.68	57.96	57.86	57.20	56.55	58.80	60.58	58.99	58.71	62.65	61.90
	Deterministic	B1	74.28	51.24	52.60	52.60	51.92	51.24	52.60	51.92	53.96	56.00	55.32
		B2	74.41	53.28	53.96	51.88	53.96	56.00	58.72	58.72	56.68	57.36	60.08
		B3	71.30	53.28	53.28	56.68	55.32	57.36	56.00	56.68	57.36	60.76	62.12

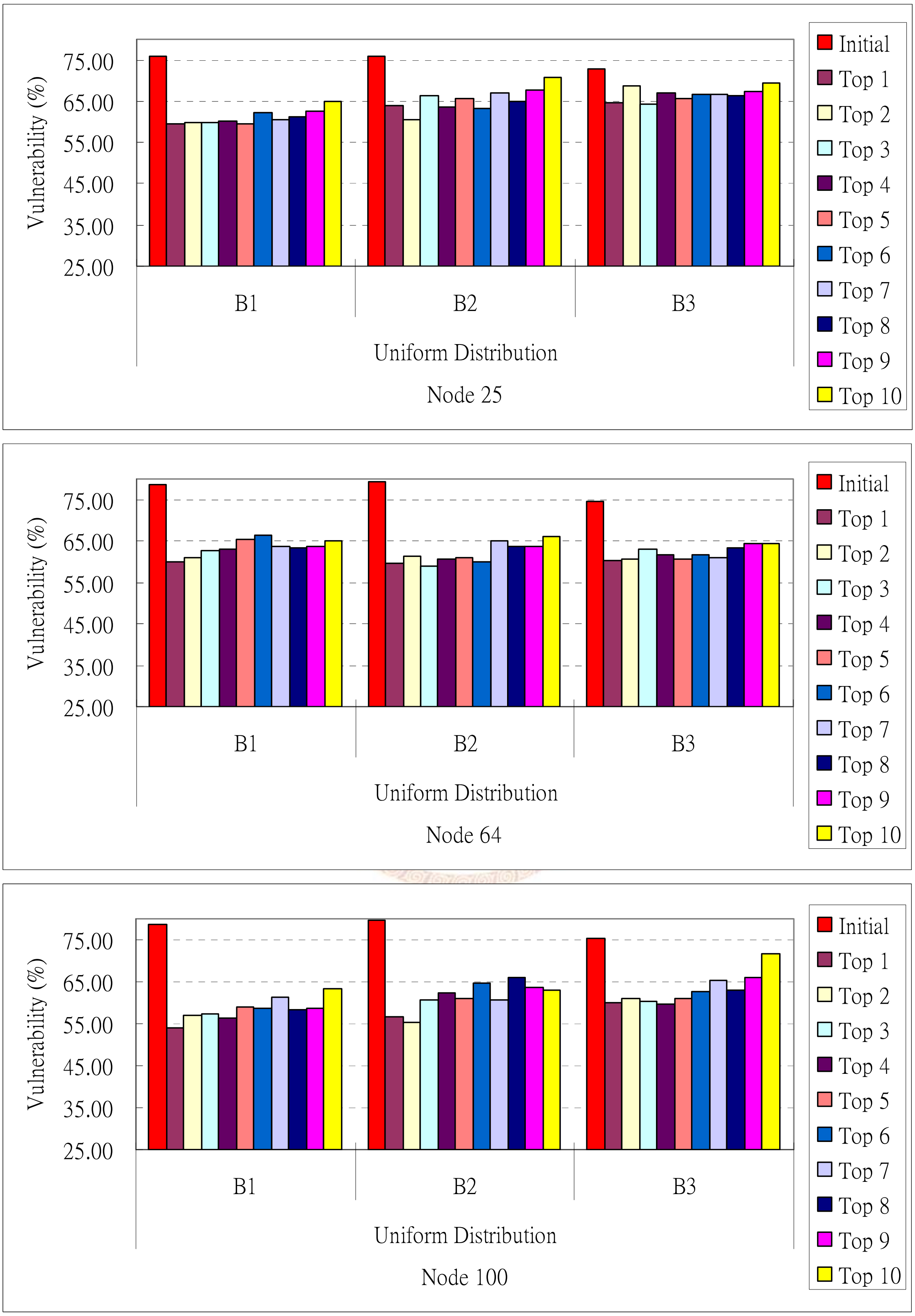


Figure 4-10 The Improvement of Top 10 Solutions under Different Network Sizes

(Uniform Distribution)

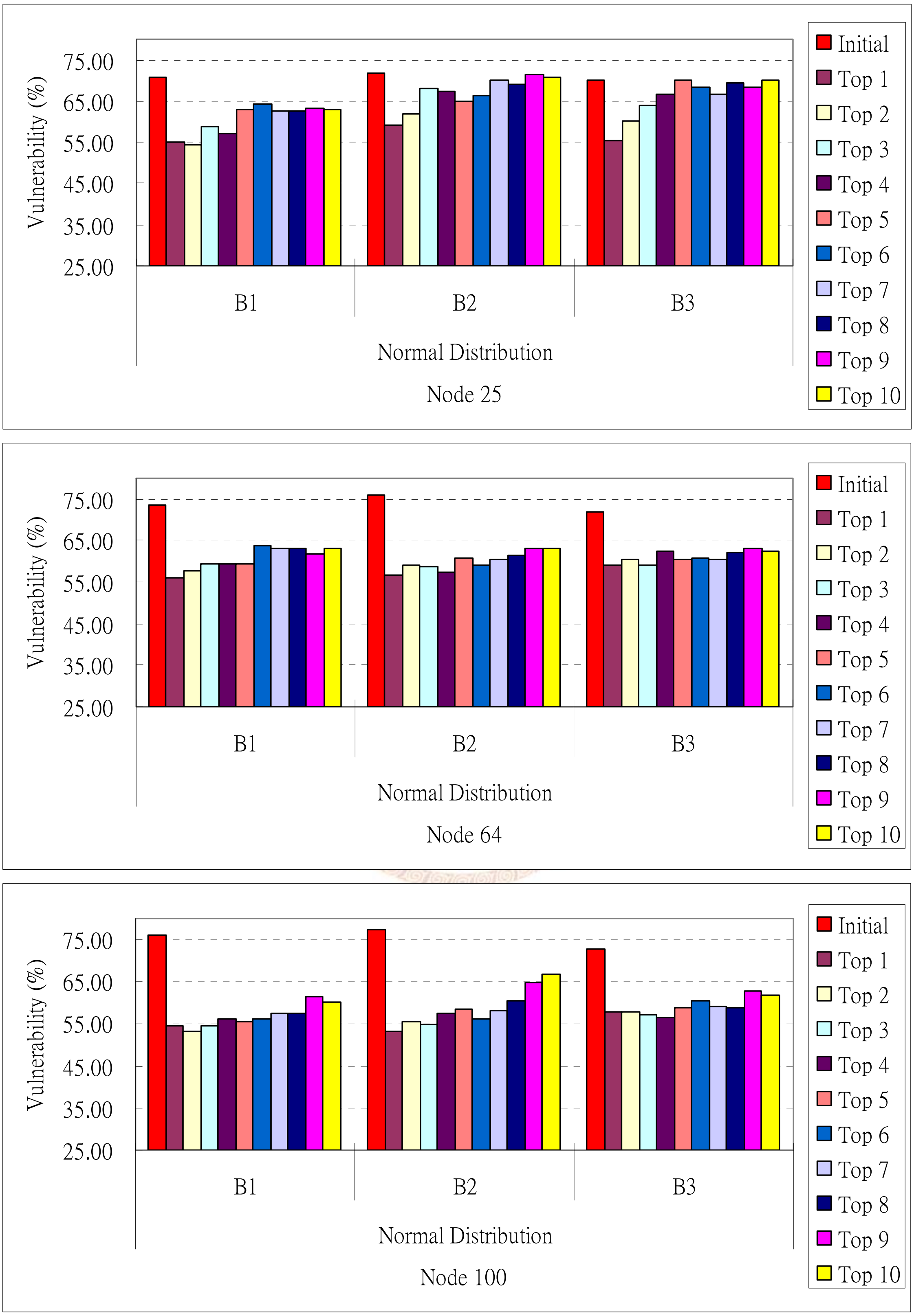


Figure 4-11 The Improvement of Top 10 Solutions under Different Network Sizes

(Normal Distribution)

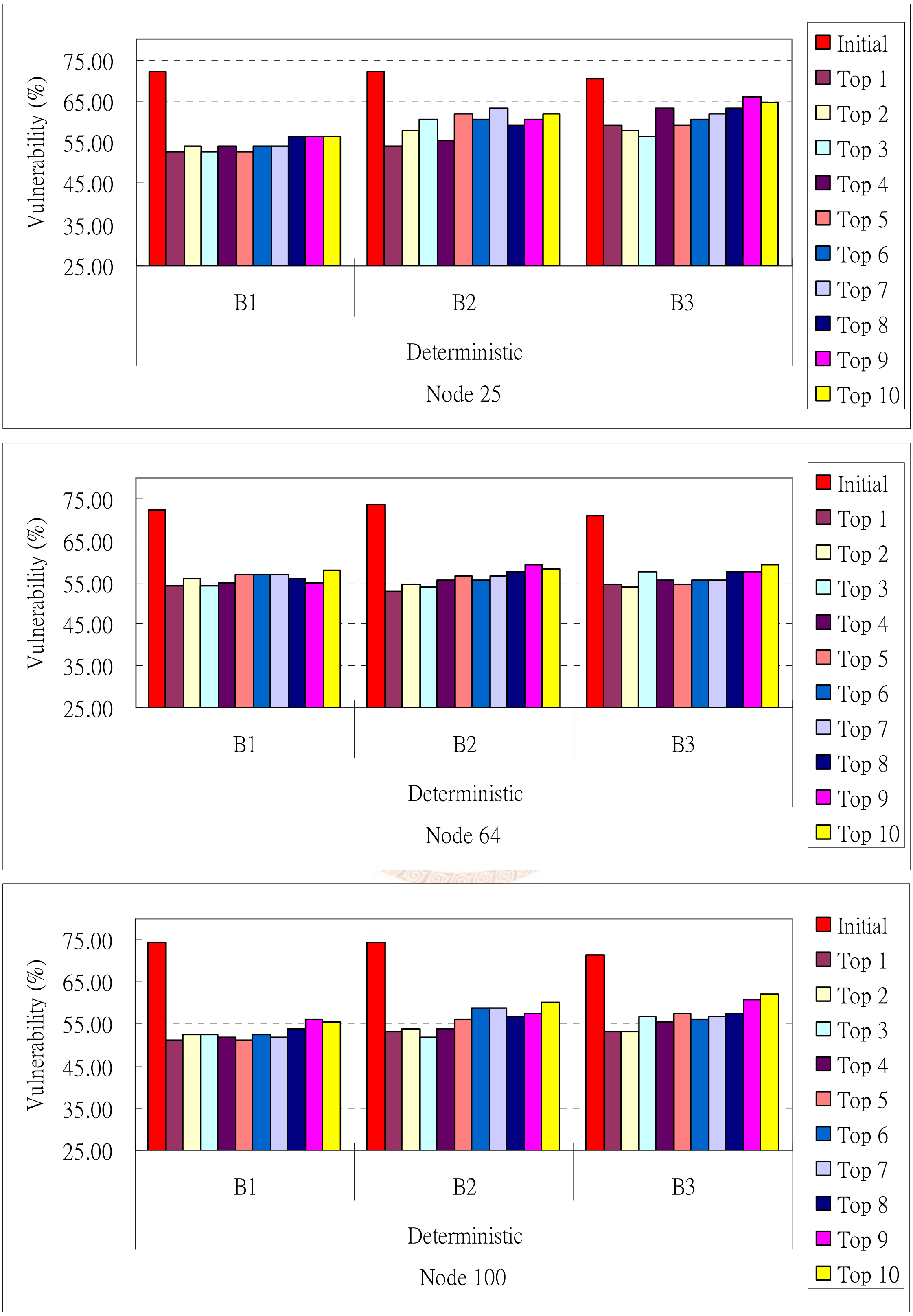


Figure 4-12 The Improvement of Top 10 Solutions under Different Network Sizes (Deterministic)

4.3.3 Discussion of Results

Figure 4-9 to Figure 4-11 display the improved vulnerability of the targeted networks under different number of nodes, damage value distribution, and the top 10 network deployment. We can observe some different findings from the original solution.

- In our original solution, the share-count based (B3) defense resource allocation strategy performs the lowest vulnerability than other strategies. However, the significant predominance of B3 strategy is eliminated after we apply the DDS model. The probable reason is that the share distribution pattern is changed so that the B3 strategy is no more appropriate in the initial network deployment.
- Especially to deserve to be mentioned, the B1 strategy is obtained relatively great improvement instead. This finding indicates the enhancement of discrete degree of secrets, including both the separation degree of shares and the difference of shares patterns among nodes. The consequence forces the attacker to make more effort to compromise more nodes in order to recover the secret. Therefore, the probability of chosen the target for each node increases so that the B1 strategy can consume attacker's power efficiently.
- In most scenarios, we observe that the average improvement ratio increase with the growth of network size generally. This phenomenon is more obvious with the B1 strategy. Since more candidate nodes can be chosen to place the shares and keys in large-sized networks, network operators enable to further enhance discrete degree.

The average improvement ratio of DDS model to initial vulnerability value under the B3 strategy is 18.65%, and it is about 22.37% in the NPDS model. Therefore, the adjustment mechanism of NPDS model is proved to be the better defense strategy since we adjust the network deployment according to the behavior of the attacker.

Chapter 5 Conclusion and Future Work

5.1 Conclusion

Because of the popularity and variety of network application, most of enterprises start to run global commerce to earn more profits by means of Internet. Therefore, the global trend leads to generating an amount of data, including the core competition and other business secrets. It is an important issue for network operators not only to guarantee QoS requirements (ex: Timely use) for subsidiaries but also to prevent information leakage from their opponents. However, there is no single mechanism of network security could solve all security threats and natural risks. A solution for security threats is to adopt several mechanisms to enhance the strength of defense. In our thesis, we address on information leakage topic as the attack behavior, hence we assist network operators to construct the robust network topology which satisfies QoS constraints for users and minimize the vulnerability caused by information leakage.

The main contribution of our research is to characterize complicated attack behaviors and real-world network strategies through the mathematical programming models, called ATSS and NPDS model. The solution approach is proposed Lagrangean Relaxation method, and we apply LR-based heuristics to solve the ATSS problem. In addition, we obtain the clues according to LR procedure and exploit our heuristics to find the near optimal solution for the NPDS problem until the attack and defense adjustments reach equilibrium. Most importantly, the result provides network operators the suggestion that how to strengthen the robustness of the network considering both the network vulnerability and network reliability.

The second contribution is to consider the random error of links in order to be close to real-world scenarios. For the sake of planning the reliable system, we must find several link disjoint paths with the concept of artificial flows and then ensure the network connectivity to reach reliability requirements of each OD pair. Moreover, we evaluate the performance under different QoS constraints that represent different ratio of additional budget extract from defense resource to maintain network reliability.

The third contribution of our research is to depend on the LR-based approach to develop defense strategy heuristics for building a secure and fault tolerant data storage service in collaborative environments. Network operators achieve the tradeoff between the confidentiality and availability with secret sharing and replication mechanisms. The concept of defense-in-depth is considered in our thesis, and the network vulnerability is improved as a whole in terms of the holistic view. Through Section 4.2, we can induce multiple defense mechanisms true more efficient than single defense mechanism. The better defense resource reallocation based on the concept that is to enhance important nodes to reach enough defense capability level rather than to strengthen the certain node extremely if the discrete degree of the shares and decrypted keys is superior.

The behavior of the attacker is also discussed the vulnerability against information leakage under different defense resource allocation strategies and several damage value patterns in our study. The experiment results indicate that the attacker trends to reveal valuable secrets in the targeted networks so that the uniform distribution is the most susceptible, while the deterministic is the least susceptible. Moreover, we could infer conclusion from the enlargement of vulnerability with the growth of number of users. That is the more number of users in our systems would make the network topology form be close to the partial mesh network because the more reliability for O-D pairs must be satisfied.

From our experiment results, we could draw some conclusions below:

- The guideline of shares and decrypted key distribution strategies are to increase the separation of shares in terms of the single secret and to differentiate the share patterns among nodes. In addition, the effect of discrete degree of secrets will be significant if all secrets in the target network are equally important.

- The rich connectivity of nodes could benefit the convenience for attackers. The discipline of topology adjustment is to set the average degree of each node to the least and similar numbers rather than form of rendezvous points according to the concept of defense-in-depth. As a result, the attacker must pay more effort to compromise the targeted node.
- The best defense resource strategy is to apply the share-count based allocation to nodal defense capability within the initial period because of protecting those nodes with more shares and keys; moreover, we must further adjust redundant defense resource to each compromised node with the uniform resource reallocation strategy. Briefly, the purpose is to allocate defense resource on relatively attractive nodes for the attackers in order to consume their attack power effectively.

5.2 Future Work

There are several research issues and conceptions to be further discussed as the following.

- **Proactive secret sharing scheme**

  In our attack-defense scenarios, we assume the attacker collects shares and decrypted keys to reveal secrets without any time limitation. However, the protection mechanism with traditional secret sharing would be insufficient for those long-lived and sensitive secrets in fact. In order to enhance security farther, the proactive secret sharing scheme is advocated [10], where shares and decrypted keys are renewed periodically without changing the secrets. In such way, the previous shares, which attackers already obtain, will become obsolete and useless after the shares are refreshed. The property of proactive approach divided all lifetime of secrets into periods of time (ex: one week, one month, etc.) and will be difficult for the attacker to recover secrets in a single time period.

- **The different attack types**

  In our thesis, we address the attack behavior of information leakage, but there are other attack types in real world, such as distributed denial of service (DDoS). The objective of attackers might be to corrupt shares resulting in information ruin or to violate QoS requirements for legitimate users. If we determine the strict threshold of the secret, it will benefit attackers to offense easily that the destruction of the secrets only requires ruining $n - k$ shares. It is a new tradeoff issue between data availability and confidentiality. Therefore, we consider various attack types and readjust our strategies to reduce the average system damage as possible in the future.

- **The better strategies for initial solutions**

  During our computational experiment phase, we can observe the initial network deployment is highly vulnerable to malicious attacks. The reason is that we merely satisfy users QoS constraints to distribute shares and decrypted keys but ignore the impact of the attacker in the initial outer solution. In other words, we only consider the availability instead of neglecting the confidentiality issue. Here, we can consider how many the numbers of shares are divided in order to observe the appropriate ratio of the total number shares to threshold for the network vulnerability. Therefore, we can further extract some constraints from the NPDS model to define the independent initial problem. If we enable to formulate the new metric which implies the difficult degree of secrets recovery for the attacker and depth-in defense for the network operator, the better initial solutions are obtained than before. For instance, we think the discrete degree of secrets as our objective function and then propose the appropriate approach to solve it such as section 3.3. The outcome is the input of the ATSS model, and it may assist the network operator to find the nearer optimal defense strategy in the NPDS model.

- **Discussion of other network planning decision variables**

  In the NPDS model, the transmission delay between arbitrary pair of nodes is the given parameter. In fact, the transmission bandwidth issue usually must be decided by network operators for network optimization. Besides, to attain the continuity of services, we can also invest redundant devices on those important nodes to reduce the possibility of security threat. To sum up, we must further take a number of QoS factors into consideration in terms of network planning. The more supplements will make our models more flexible and closer to the real world.

References

[1] D.A. Grier, “The Innovation Curve,” *IEEE Computer Society*, Vol. 39, pp. 8-10, February 2006.

[2] A. Azadmanesh, A.W. Krings, and P.W. Oman, “Security and Survivability of Networked Systems,” *Proceedings of the 38th IEEE Hawaii International Conference on System Sciences*, 2005.

[3] R. Richardson, “2008 CSI Computer Crime and Security Survey,” *Computer Security Institute*, 2008, http://www.gocsi.com/.

[4] M. Al-Kuwaiti, N. Kyriakopoulos, and S. Hussein, “Network Dependability, Fault-tolerance, Reliability, Security, Survivability: A Framework for Comparative Analysis,” *The George Washington University*, November 2006.

[5] G.R. Ganger, H. Kılıççöte, J.D. Strunk, J.J. Wylie, M.W. Bigrigg, and P.K. Khosla, “Survivable Information Storage Systems,” *Garnegie Mellon University*, August 2000.

[6] “Information Security in Taiwan,” http://www.informationsecurity.com.tw.

[7] G. Levitin, “Optimal Defense Strategy against Intentional Attacks,” *IEEE Transactions on Reliability*, Vol. 56, No. 1, March 2007.

[8] A. Shamir, “How to Share a Secret,” *Massachusetts Institute of Technology*, 1979.

[9] M. Tompa, “How to Share a Secret with Cheaters,” *International Association for Cryptologic Research*, 1988.

[10] A. Herzberg, H. Krawczyk, M. Yung, and S. Jarecki, “Proactive Secret Sharing or How to Cope with Perpetual Leakage,” *IBM T.J. Watson Research Center*, 1995.

[11] A. Subbiah and D.M. Blough, “An Approach for Fault Tolerant and Secure Data Storage in Collaborative Work Environments,” *School of Electrical and Computer*

Engineering Georgia Institute of Technology, April 2005.

[12] B. Wang and J.C. Hou, "Multicast Routing and Its QoS Extension: Problems, Algorithms, and Protocols," *The Ohio State University*.

[13] J. Crowcroft and Z. Wang, "Quality of Service Routing for Supporting Multimedia Applications," *IEEE Journal on Selected Areas in Communications*, September 1996.

[14] K. Nahrstedt and S. Chen, "An Overview of Quality-of-Service Routing for the Next Generation High-Speed Network: Problems and Solutions," *IEEE Network*, December 1998.

[15] G. Karlsson and I. Mas, "Quality of Service and the End to End Argument," *IEEE Network*, November 1997.

[16] D.A. Fisher and H.F. Lipson, "Survivability - A New Technical and Business Perspective on Security," *Coordination Center Software Engineering Institute*, 2000.

[17] D.A. Fisher, H.F. Lipson, N.R. Mead, R.C. Linger, R.J. Ellison, and T.A. Longstaff, "Survivable Network Systems: An Emerging Discipline," *Technical Report CMU/SEI-97-TR-013, Software Engineering Institute*, *Carnegie Mellon University*, pp. 1-31, November 1997 (Revised 1999).

[18] A. Krings, "Design for Survivability: A Tradeoff Space," *ACM International Conference Proceeding Series*, Vol. 288, 2008.

[19] J.L. Tzeng, "Near Optimal Network Defense Resource Allocation Strategies for the Minimization of Information Leakage," *Department of Information Management, National Taiwan University*, 2006.

[20] M.N. Azaiez and V.M. Bier, "Optimal Resource Allocation for Security in Reliability Systems," *European Journal of Operational Research*, Vol. 181, No. 2,

pp. 773-786, September 2007.

[21] A.M. Geoffrion, “Lagrangean Relaxation and Its Use in Integer Programming,” *Mathematical Programming Study*, Vol. 2, pp. 82-114, 1974.

[22] M.L. Fisher, “The Lagrangian Relaxation Method for Solving Integer Programming Problems,” *Management Science*, Vol. 27, No. 1, pp. 1-18, January 1981.

[23] M.L. Fisher, “An Application Oriented Guide to Lagrangean Relaxation,” *Interfaces*, Vol. 15, No. 2, pp. 10-21, April 1985.

[24] C.D. Gelatt, M.P. Vecchi, and S. Kirkpatrick, “Optimization by Simulated Annealing,” *Science*, Vol. 220, No. 4598, pp. 671-680, May 1983.

簡歷

姓名：陳冠瑋

出生地：台灣 台北市

生日：中華民國七十四年五月十六日

學歷：九十二年九月至九十六年六月
國立中央大學 資訊管理學系學士

九十六年九月至九十八年七月
國立台灣大學 資訊管理學研究所碩士