# Near Optimal Secret Sharing for Information Leakage Maximization

Frank Yeong-Sung Lin, Kuo-Chung Chu, Pei-Yu Chen, and Guan-Wei Chen

Department of Information Management National Taiwan University
Taipei, Taiwan, R.O.C.
yslin@im.ntu.edu.tw, kcchu@ntcn.edu.tw, d96006@im.ntu.edu.tw,
r96037@im.ntu.edu.tw

**Abstract.** In this paper, we propose a mathematical programming model to describe an offense-defense scenario. In the offense problem, the objective of attackers is to compromise nodes in order to steal information. Therefore, the attackers try to recover secrets through compromising certain nodes and to maximize the information leakage as much as possible. During the attack actions, the attacker must allocate a limited budget to collect a large enough number of shares and decrypted keys through compromising certain nodes. Therefore, we advocate Lagrangean Relaxation algorithms and the proposed heuristics to find a near optimal solution. Through solutions from the perspective of the attacker, we then induce some efficient defense mechanisms for the network operators.

**Keywords:** Optimization, Lagrangean Relaxation, Resource Allocation, Network Planning, Secret Sharing, Information Security, Reliability, Survivability.

## 1 Introduction

The rapid growth of the Internet has made many individuals, schools and enterprises generate a great deal of demands, and how to share the information has securely become an important issue. However, rapid upgrades in technology still bring a negative side effect: computer crimes also increase rapidly [1]. Hackers engage in cyber crime by applying a variety of tools, such as injecting worms or using backdoor programs for web phishing, to steal information for fun or gaining benefits, . These cyber-crime events have already been urgent problems for a while and have caused serious damage to network security, especially regarding information leakage. In fact, sometimes the cyber crimes happen without anyone noticing until attackers announce or publicize the stolen information. Typically, an information system is a "Survivable Storage System" and should provide continuous service for each legitimate user even if the system suffers from intentional attacks or natural accidents. People access necessary data and information through digital storages frequently so that the confidentiality, reliability, availability, integrity of the data storage devices must be considered as a very important aspect of information security.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part III, LNAI 6098, pp. 189–198, 2010.

In this manner, the combinative method of secret sharing [3] and replication mechanisms [4] achieves the goal of information security, and it provides users with the ability of increasing degrees of security to store critical information to ensure its persistence, continuous accessibility, indestructibility, and confidentiality [5].However, there is a tradeoff between the depth and the width of deployment to process the secret sharing scheme, such as the greater system security would result in the less availability of information when it is needed. Whoever can obtain a large enough number of shares and the corresponding decrypted key may recover the secret.

In an offense scenario, assuming the attacker is extremely skilled, he will always find the most efficient way to maximize system damage. For example, he will know which node contains what kind of shares and decrypted keys, and he will consider the nodal capability and the benefit on this node to decide whether to compromise it or not. If the attacker gets more than the threshold number of shares, he can cause serious damage to both reputation and profit. To access more information, the attacker must choose the best ratio of the attack cost to benefit gained because the budget is finite. Even though a lot of drawbacks and risks exist in networks, most enterprises still store and share data and information by means of the network system. Some problems, for example, derive from information backups, recoveries, and the sharing of information between legitimate users, while we deal with 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 [6]. However, there are few researches that discuss system performance and the offense-defense scenarios in mathematical ways. Therefore, we propose a mathematical model, called the Attack Target Selecting Strategy (ATSS). The objective of this model is to recover the secret information and maximize the total damage.

The remainder of this paper is organized as follows. In next section, a mathematical formulation of the offense-defense scenario is proposed. The proposed solution approach which is based on the Lagrangean Relaxation method is presented after the problem formulation. Then we discuss the computational results and implications. Finally, the last section discusses conclusions and future works from this research.

## 2 Problem Formulation

Business enterprises have a lot of sensitive information and business secrets, such as customer data or core knowledge. If the information leakage occurs, the victims can lose competitive advantages and perhaps even lead to loss of reputation. Therefore, enterprises must adopt multiple mechanisms to reduce the impact of information leakage. One such mechanism is that information should be encrypted before applying a secret sharing scheme so as to further enhance confidentiality. In this case, the attacker and users must obtain enough shares and decrypted keys if they want to recover the secret to the defense capability to avoid a waste of attack cost, to compromise a node and cause information leakage,

The network we discussed is an Autonomous-System (AS) level Internet. Topology is an undirected graph, in which the nodes are connected by an undirected link, and each node represents a domain and each link represents the inter-domain connection. The attacker outside the AS must enter into the AS through compromising the

entry node and compromise nodes step-by-step in the targeted network. 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.

### 2.1 Problem Formulation of the ATSS Model

The problem of attacker behavior is formulated as a mathematical programming problem. The given parameters and decision variables are defined in Table 1 and Table 2, respectively.

**Table 1.** 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 $w \in W$ |
| $\delta_{pl}$ | The indicator function, which is 1 if the link or the node *l* is on path *p*, and 0 otherwise (where $l \in N_1 \cup L_1,\ p \in p_w$) |
| $\Omega_l$ | 1 if link l is selected to implement, and 0 otherwise (where $l \in L_1$) |
| $\mathcal{V}$ | The index set of all sensitive information |
| $m_\upsilon$ | The share index of the secret $\upsilon$, where $\upsilon \in \mathcal{V}$ |
| $\alpha_{im\upsilon}$ | 1 if the node *i* stores shares of index *m*, and 0 otherwise (where $i \in N_1, m \in m_\upsilon, \upsilon \in \mathcal{V}$) |
| $\eta_{i\upsilon}$ | 1 if the node *i* stores the decrypted key of the secret $\upsilon$, and 0 otherwise (where $i \in N_1, \upsilon \in \mathcal{V}$) |
| $S_\upsilon$ | Damage incurred by leaking at least $k_\upsilon$ pieces of the secret $\upsilon$ and the corresponding decrypted key, where $\upsilon \in \mathcal{V}$ |
| $k_\upsilon$ | The threshold number of shares required to recover the secret $\upsilon$, where $\upsilon \in \mathcal{V}$ |
| $\tilde{a}_i(b_i)$ | 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.** Decision Variables

| Notation | Description |
|---|---|
| $a_i$ | The attack budget allocated to compromise the node, where $i \in N_1$ |
| $Z_\upsilon$ | 1 if both $k_\upsilon$ shares and the key are stolen and 0 otherwise (where $\upsilon \in \mathcal{V}$) |
| $x_p$ | 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 $i \in N_1$) |

The objective of an attacker is to maximize damage through compromising nodes and then obtaining the essential shares. In this problem, the attacker must use a finite budget to recover maximal damage in the targeted network. Hence, we model this problem as an optimization problem called the ATSS Model. The formulation is shown as follows.

**Objective function:**

$$\underset{Z_v, y_i, a_i, x_p}{Max} \ (\sum_{v \in \nu} S_v Z_v), \qquad \text{IP 1}$$

**Subject to:**

$$\sum_{p \in P_w} x_p = y_i \qquad \forall i \in N_1, w = (o,i) \qquad \text{IP 1.1}$$

$$\sum_{p \in P_w} x_p \le 1 \qquad \forall w \in W \qquad \text{IP 1.2}$$

$$x_p = 0 \ or \ 1 \qquad \forall p \in P_w, w \in W \qquad \text{IP 1.3}$$

$$y_i = 0 \ or \ 1 \qquad \forall i \in N_1 \qquad \text{IP 1.4}$$

$$\sum_{w \in W} \sum_{p \in P_w} x_p \delta_{pi} \le (|N_1| - 1) y_i \qquad \forall i \in N_1 \qquad \text{IP 1.5}$$

$$\sum_{p \in P_w} x_p \delta_{pl} \le \Omega_l \qquad \forall l \in L_1, w \in W \qquad \text{IP 1.6}$$

$$\sum_{i \in N_1} a_i \le A \qquad \text{IP 1.7}$$

$$0 \le a_i \le \hat{a}_i(b_i) \qquad \forall i \in N_1 \qquad \text{IP 1.8}$$

$$\hat{a}_i(b_i) y_i \le a_i \qquad \forall i \in N_1 \qquad \text{IP 1.9}$$

$$k_v Z_v \le \sum_{m \in m_v} \sum_{i \in N_1} (\alpha_{imv} y_i) \qquad \forall v \in \nu \qquad \text{IP 1.10}$$

$$Z_v \le \sum_{i \in N_1} \eta_{iv} y_i \qquad \forall v \in \nu \qquad \text{IP 1.11}$$

$$Z_v = 0 \ or \ 1 \qquad \forall v \in \nu. \qquad \text{IP 1.12}$$

The attacker chooses the attack path to reach the target *i*, and the intermediate nodes that are all compromised are determined in Constraint (IP 1.1) to Constraint (IP 1.5). Constraint (IP 1.6) restricts the attack path *p* which must be constructed on the implemented link. The attacker thus allocates his budget to compromise the node. And Constraint (IP 1.7) to Constraint (IP 1.9) are constrains about the allocated budget to compromise the node of attackers. Constraint (IP 1.10) to Constraint (IP 1.12) jointly enforce that the attacker doesn't cause the information damage $S_u$ unless he gets the decrypted key and reveals the threshold $k_u$ of shares by compromising the nodes.

## 3 Solution Approach

### 3.1 The Lagrangean Relaxation-Based Algorithm

In this section, the Lagrangean Relaxation method [7] is applied to solve the ATSS model. First, complicating constraints are relaxed by being multiplied with the corresponding Lagrangean multipliers, the product of which is then added to the primal objective function, and then the LR problem is generated. Second, the LR problem is decomposed into four subproblems according to the decision variables. Each subproblem adopts the well-known algorithm to solve it optimally and easily. More detailed procedures about the proposed model are described in the following sections.

### 3.2 The Lagrangean Relaxation Problem

In the ATSS Model, we relax Constraint (IP 1.1), (IP 1.5), (IP 1.9), (IP 1.10), and (IP 1.11). The model is transformed to an optimization (LR 1). More details about (LR 1) is shown below.

**Optimization problem:**

$$\begin{aligned} Z_D(\mu_1,\mu_2,\mu_3,\mu_4,\mu_5) = \min_{Z_v, y_i, a_i, x_p} & -\sum_{v\in V} S_v Z_v + \sum_{i\in N_1} \mu_i^1 \{ \sum_{p\in P(o,i)} x_p - y_i \} \\ & + \sum_{i\in N_1} \mu_i^2 \{ \sum_{w\in W} \sum_{p\in P_w} x_p \delta_{pi} - (|N_1|-1) y_i \} + \sum_{i\in N_1} \mu_i^3 \{ \hat{a}_i(b_i) y_i - a_i \} \\ & + \sum_{v\in V} \mu_v^4 \{ k_v Z_v - \sum_{m\in m_v} \sum_{i\in N_1} \alpha_{imv} y_i \} + \sum_{v\in V} \mu_v^5 \{ Z_v - \sum_{i\in N_1} \eta_{iv} y_i \}, \end{aligned} \quad \text{LR 1}$$

Subject to (IP 1.2) to (IP 1.4), (IP 1.6) to (IP 1.8)and (IP 1.11) to (IP 1.12)

**Subproblem 1 (related to decision variable $x_p$)**

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

**Subject to**(IP 1.2), (IP 1.3) and (IP 1.6)

In subproblem 1, we arrange the problem (Sub 1) as Equation (1).

$$Z_{Sub\,1}(\mu_1,\mu_2) = \min \sum_{w\in W} \sum_{p\in P_w} [ \sum_{j\in N_1} \mu_j^2 \delta_{pj} + \mu_i^1 ] x_p . \quad (1)$$

To reduce the complexity, subproblem 1 is then decomposed in into |*W*| problems, which are all independent shortest path problems. We individually determine the value of $x_p$ for each O-D pair *w*. In this problem, Dijkstra's minimum cost shortest path algorithm is applied the to solve it, and time complexity is $O(|N_1|^2)$.

**Subproblem 2 (related to decision variable $Z_u$)**

$$Z_{Sub\,2}(\mu_4,\mu_5) = \min\{ -\sum_{v\in V} S_v Z_v + \sum_{v\in V} \mu_v^4 k_v Z_v + \sum_{v\in V} \mu_v^5 Z_v \}, \quad \text{Sub 2}$$

**Subject to** (IP 1.12)

In Subproblem 2, we arrange (Sub 2) as Equation (2), and decompose it into |*v*| independent subproblems, where we decide the value of $Z_u$ of the secret *u*.

$$Z_{Sub\,2}(\mu_4,\mu_5) = \min \sum_{v\in V} (-S_v + \mu_v^4 k_v + \mu_v^5) Z_v . \quad (2)$$

If $(-S_v + \mu_v^4 k_v + \mu_v^5)$ is non-positive, the value of $Z_u$ must be set to one for each sensitive information because of the minimum problem, and zero otherwise. The time complexity of (Sub 2) is $O(|v|)$.

**Subproblem 3 (related to decision variable $y_i$ )**

$$\begin{aligned} & Z_{Sub\,3}(\mu_1,\mu_2,\mu_3,\mu_4,\mu_5) \\ & = \min \sum_{i\in N_1} \mu_i^1 (-y_i) + \sum_{i\in N_1} -\mu_i^2 (|N_1|-1) y_i ) + \sum_{i\in N_1} \mu_i^3 (\hat{a}_i(b_i) y_i ) \\ & + \sum_{v\in V} \sum_{m\in m_v} \sum_{i\in N_1} (-\mu_v^4 \cdot \alpha_{imv} y_i ) + \sum_{v\in V} \sum_{i\in N_1} -\mu_v^5 \eta_{iv} y_i , \end{aligned} \quad \text{Sub 3}$$

**Subject to** (IP 1.1)

The same concept is presented above, and accordingly the (Sub 3) is reformed as Equation (3).

$$Z_{\text{Sub 3}}(\mu_1,\mu_2,\mu_3,\mu_4,\mu_5) = \min \sum_{i\in N_1}\{-\mu_i^1 - \mu_i^2(|N_1|-1) + \mu_i^3(\hat{a}_i(b_i)) + \sum_{v\in V}(\sum_{m\in m_v}(-\mu_v^4\alpha_{imv}) - \mu_v^5\eta_{iv})\}y_i. \quad (3)$$

Then we can further decompose $Z_{\text{Sub 3}}$ in Equation (3) into $|N_1|$ independent subproblems. We must determine the value of $y_i$ of the actual node $i \in N_1$. Since this is a minimum problem, we set one if the coefficient of $y_i$ are non-positive, zero otherwise. The exhausting search algorithm is applied to solve this subproblem. The time complexity of (Sub 3) is $O(|N_1||v||m_u|)$.

**Subproblem 4 (related to decision variable $a_i$)**

$$Z_{\text{Sub 4}}(\mu_3) = \min \sum_{i\in N_1} -\mu_i^3 a_i, \quad \text{Sub 4}$$

**Subject to**(IP 1.7) and (IP 1.8)

Although (Sub 4) traditionally minimizes negative loss rather than maximizes positive profit, it can be viewed as a fractional knapsack problem. First, we use the parameter $-\mu_i^3$ as the weight of the artificial link, and then sort each actual node $i \in N_1$ by weight in ascending order. Second, we allocate the value of $a_i$ to $a_i(b_i)$ from the smallest $-\mu_i^3$ to the biggest one until the sum of allocated $a_i$ equals or exceeds $A$; if the last $a_i$ is insufficient to set $a_i(b_i)$, the last $a_i$ is set to the value of the sum of $a_i$ subtracted from $A$. The time complexity of this problem is $O(|N_1|^2)$.

### 3.3 Getting Primal Feasible Solutions

We get the lower bound (LB) of the primal problem after solving the LR problem and further use a multiplier to adjust the original algorithm to the LR-based modified heuristic algorithm to obtain the upper bound (UB). We try to derive the tightest gap between the UB and LB with the proposed heuristic, so we iteratively use the subgradient technique to adjust the multipliers as good as possible. The basic concept of the LR-based heuristic we propose here is to extract $X_p$'s value from the subproblem as the candidate attack path and compromise all nodes on these paths. Here, we calculate the node weight, which reflects the ratio of the attack cost to the profit gained. The smaller the node weight is, the more attractive it is to attackers. The node weight is recalculated continually according to the condition of secrets. This mechanism assists the attacker to cause information leakage more effectively.

If attack costs exceed budget, the attacker chooses the largest node weight to remove its attack cost until the attack cost is feasible. The objective of the procedure is to reserve the maximal profit on this attack tree and to abandon relatively worthless nodes. On the other hand, if the attack cost is smaller than budget, it means the attacker still has some budget to allocate. However, what differs from before is that we set the compromised nodes' weight to zero here. The resulting node weight is considered the cost to implement Prim's Algorithm, and then each node's path weight from the attacker's initial position can be known. For each unrecovered secret, we sum up the weight of the paths until the secret can be recovered, and set the smallest weight to be the targeted secret. Therefore, the attacker will attack uncompromised nodes on

chosen paths to recover the targeted secret. The procedure for choosing the to-be-recovered 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.

# 4 Computation Experiments

## 4.1 Experiment Environment

To measure the effect of different damage value distributions for the attacker, we design three different patterns of damage value in this paper. The first is uniform distribution, which is the scope of the information value, 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 distribution, whose secret damage is the same, meaning each secret is equally important.

We also design the different number of users to measure the vulnerability. The greater the number of legitimate users that exist, the greater the reliability network operators must guarantee. Furthermore, there are three budget allocation strategies, denoted B1, B2 and B3. B1 is a uniform-based budget allocation, where for each node we allocate the same defense budget. B2 is a degree-based budget allocation, where we allocate the defense budget according to the percentage that the degree number of the node is over the total degree of the network. B3 is a share-count-based budget allocation, where we allocate the defense budget depending on how many shares and decrypted keys the node contains. The network size consists of 25, 64, 100 nodes in each scenario. The defense capability function is defined as a concave form, $2 \log(6b_i+1)+\varepsilon$, where $b_i$ is the budget allocated to node $i$.

## 4.2 Computation Experiments with the ATSS Model

To evaluate whether the performance of our proposed heuristic algorithm is effective we implement two simple algorithms for comparisons, denoted SA1 and SA2. The concept of SA1 depends on the current condition of the secret to determine which node with the smallest weight has the highest priority to be compromised. SA1 creates the Next_Attack_candidate to record the set of the neighborhood of the attack tree, and all nodes in the Next_Attack_candidate are the candidate targets. After setting the weight of the node, we apply the greedy algorithm to construct an attack tree from the attacker's initial position. The procedure is repeated until the attack budget is exhausted. The total computational complexity of the is $O(|N_I|^3)$. Briefly, the main idea of SA1 arises from the intention of the attacker to compromise nodes with the smallest weight to obtain the most beneficial effect.

The fundamental concept of SA2 is derived from the LR-based heuristic. First, the sum of paths with the smallest weight is chosen, which is set as the target to be recovered in terms of the state of all secrets. Second, for each unrecovered secret, we sum the weight of the first $k_u$ *th* path, whose nodes contain shares and the decrypted key, and then sort the unrecovered secrets by the weights in ascending order. Third, we set the secret with the smallest weight of the sum of paths as the target, meaning that it obtains the most beneficial effect. Finally, the uncompromised nodes 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 SA2 is $O(|N_I||v^2||m_u|)$.The experiment result is shown in Fig. 1 to Fig. 4. The discussions and implications of the experiment are in the next section.

### 4.3 Discussion of Results

From Fig. 1 to Fig. 3, the target network with the B3 strategy is the least vulnerable and the most robust in all cases. It performs more successfully than the other strategies since the defense resource is allocated according to the importance of each node. The more shares and keys that a node contains, the higher probability of it is chosen as targets by attackers. With the growth of the network, the difference between B1, B2 and B3 increase significantly for each damage value distribution. The attacker could choose more targets so that the influence of wrong defense allocation would be magnified for network vulnerability.

The trend of network vulnerability rises if the system must provide more users with QoS requirements. Take Deterministic distribution as an example. All damage value is system reliability being achieved to a certain level; the network operator must transfer some budget from defense budget. For each damage value distribution, the decision of the attacker is equally important, so the attacker chooses the targets according to the kind of shares and obtained keys. Fig. 4 illustrates the network vulnerability of our proposed algorithm is always higher than SA1 and SA2 among damage value distributions. Clearly, the proposed heuristics provide a better solution than SA1and SA2.

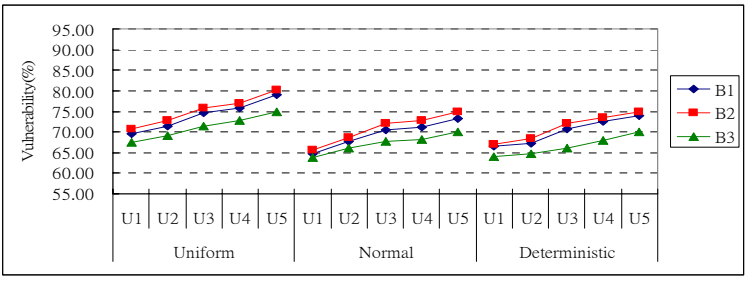


**Fig. 1.** The Network Vulnerability under Different Numbers of Users ($|N_I|$= 25)

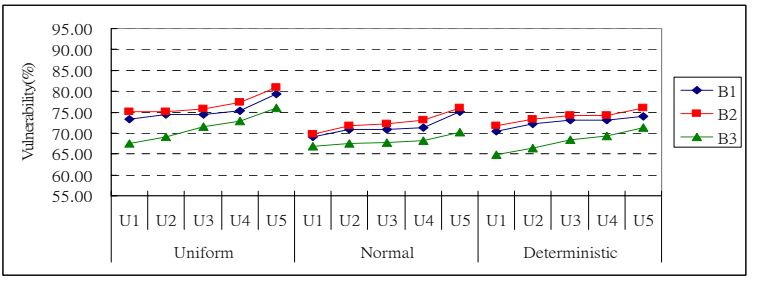


**Fig. 2.** The Network Vulnerability under Different Numbers of Users ($|N_I|$= 64)

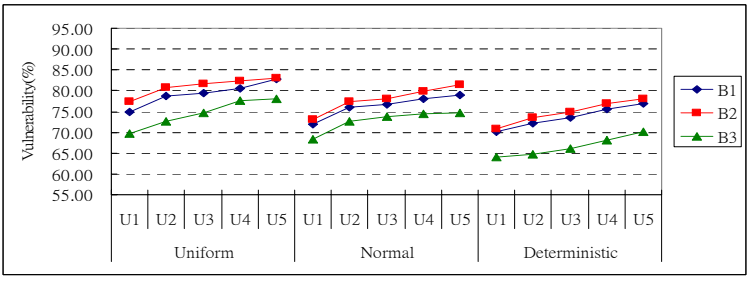


**Fig. 3.** The Network Vulnerability under Different Numbers of Users ($|N_1|$= 100)

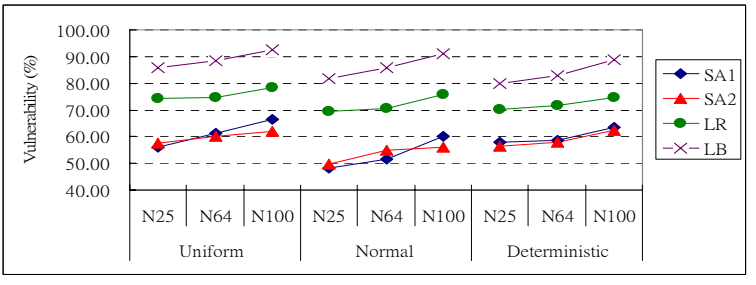


**Fig. 4.** Vulnerability of Different Network Sizes and Damage Distributions

## 5 Conclusion

In our paper, we assist defenders by constructing a robust network topology for users and minimize the vulnerability caused by information leakage. From the experiment results, the guideline of shares and decrypted keys distribution strategies increase the separation of shares in terms of the single secret and to differentiate the share patterns among nodes; the discipline of topology adjustment is to set the average degree of each node to the least and the same numbers rather than form of rendezvous points considering the concept of defense-in-depth; the best strategy is to allocate resources on relatively attractive nodes for attackers.

The main contribution of this paper is to characterize complicated attack behaviors and real-world network strategies through mathematical models. In addition, we obtain clues from Lagrangean Relaxation procedures and exploit our heuristics to find a near optimal solution. Network operators can achieve the tradeoff between the confidentiality and availability with secret sharing and replication mechanisms. The concept of defense-in-depth is considered in our paper so that the network vulnerability is reduced as a whole in terms of a holistic view. Through the experiment results, we could induce multiple defense mechanisms that were actually more efficient than a single defense mechanism. In this paper, we assume the attacker collects shares and decrypted keys to reveal secrets without time limitation. However, this would be insufficient for evaluating the survivability. To enhance security further, the proactive secret sharing scheme is advocated [8], where shares and decrypted keys are renewed

periodically without changing the secrets. The property of the proactive approach divided all lifetimes of secrets into periods of time and will be difficult for the attacker to recover secrets in a single time period. Therefore, we intend to discuss the proactive secret sharing scheme in the future.

## References

1. Azadmanesh, A., Krings, A.W., Oman, P.W.: Security and Survivability of Networked Systems. In: Proceedings of the 38th IEEE Hawaii International Conference on System Sciences (2005)
2. Al-Kuwaiti, M., Kyriakopoulos, N., Hussein, S.: Network Dependability, Fault-tolerance, Reliability, Security, Survivability. In: A Framework for Comparative Analysis. The George Washington University (2006)
3. Shamir, A.: How to Share a Secret. Massachusetts Institute of Technology (1979)
4. Subbiah, A., Blough, D.M.: An Approach for Fault Tolerant and Secure Data Storage in Collaborative Work Environments. School of Electrical and Computer Engineering Georgia Institute of Technology (2005)
5. Ganger, G.R., Kiliççöte, H., Strunk, J.D., Wylie, J.J., Bigrigg, M.W., Khosla, P.K.: Survivable Information Storage Systems. Garnegie Mellon University (2000)
6. Levitin, G.: Optimal Defense Strategy against Intentional Attacks. IEEE Transactions on Reliability 56(1) (2007)
7. Fisher, M.L.: An Application Oriented Guide to Lagrangean Relaxation. Interfaces 15(2), 10–21 (1985)
8. Herzberg, A., Krawczyk, H., Yung, M., Jarecki, S.: Proactive Secret Sharing or How to Cope with Perpetual Leakage. IBM T.J. Watson Research Center (1995)