



ELSEVIER

Information Sciences 116 (1999) 109–127

INFORMATION
SCIENCES
AN INTERNATIONAL JOURNAL

A probability model for reconstructing secret sharing under the internet environment

Ching-Yun Lee ^{a,1}, Yi-Shiung Yeh ^{a,2}, Deng-Jyi Chen ^{a,*},
Kuo-Lung Ku ^b

^a *Institute of Computer Science and Information Engineering, National Chiao-Tung University, Hsinchu 30050, Taiwan, ROC*

^b *Chung-Shan Institute of Science and Technology, Taoyuan, Taiwan, ROC*

Received 17 January 1998; received in revised form 13 August 1998; accepted 24 October 1998

Abstract

Internet technologies have ushered in a new era for computer-related communications. Use of the Internet for commercial applications and resource sharing has accelerated in recent years as well. Owing to such developments, Internet security has become a critical issue for both academic and commercial sectors. Several studies have focused on areas involving network security such as authentication, privacy, and integrity. In some applications, an important message can be divided into pieces and allocated at several different sites over the Internet for security access concern. For example, an important map that can be used to access a military base, a vital key that can be used to give a military order or command. To access such an important message, one must reconstruct the divided pieces from different sites under the given Internet environment. In this paper, we present a novel probability model for reconstructing secret sharing under the Internet. Also, how to assign the divided shares into different sites over the Internet is studied. Also proposed herein are algorithms to perform shares assignment and to evaluate the probability of reconstructing the divided pieces into the original secret. Illustrative examples and simulation results demonstrate the feasibility of the proposed approach. © 1999 Elsevier Science Inc. All rights reserved.

Keywords: Internet; Secret sharing; Graph theory; Spanning tree; Probability

* Corresponding author. Fax: +886 3 572 4176; e-mail: djchen@csie.nctu.edu.tw

¹ E-mail: leecy@csie.nctu.edu.tw

² E-mail: ysieh@csie.nctu.edu.tw

1. Introduction

The Internet has revolutionized computer-related communications. The Internet consists of networks interconnected by a set of routers which allow them to function as a single, large virtual network. The Internet functions as a global broadcasting medium, a mechanism for information dissemination, and a forum for collaboration and interaction between individuals and their computers regardless of geographical proximity. Such a capacity accounts for why use of the Internet has exploded in recent years. According to Network Wizards³, the number of hosts on the Internet grew from 1,313,000 in January 1993 to 36,739,000 in July 1998. An added feature of the Internet is resource sharing among remote users, in which user can quickly access data from any site. Nevertheless, given its openness and easy access to the general public, the Internet is vulnerable to more attacks from intruders than any other network.

A user can achieve a secure secret in Internet environments by adopting encryption/decryption algorithms [1,2] and secure key distribution protocols [3,4] as well as secret sharing schemes [5–7]. Sharing secrets among several individuals in a manner that no individual holds all the secrets is highly desired. Shamir [8] and Blakley [9] pioneered the notion of secret sharing and provided the secret sharing schemes. An (m, n) secret sharing scheme is a method which a secret, S , is divided into n shares (or shadows) in such a manner that the secret S cannot be reclaimed unless at least m shares are collected. This scheme is commonly referred to as (m, n) -threshold scheme. Stinson [10] and Simmons [11] thoroughly review related developments on the secret sharing schemes.

A secret can be taken and divided it into pieces in several ways. To secure a secret, we can divide secret into shares and store shares at different sites over the Internet. The Internet provides resource sharing among remote users, thereby allowing quick access of a message from any site. As generally known, a user can reconstruct a secret message by sending requests to the service. A service exports a set of commands. After executing a command, the service can transfer the requested shares to the user that invoked the command. Secret sharing schemes can be used for diverse applications, including important maps, secret recipes, and military command and control systems such as a fire controller. In addition, a general sharing scenario one can imagine can be modeled.

Two types of shares, single-share and multiple-share, play a prominent role in the development of the share assignment method. Each participant could have only one share or a different number of shares (multiple-share). Single assignment secret sharing scheme assumes that each participant holds only one

³ This data is provided by Network Wizards and is available on the Internet at <http://www.nw.com/>.

share. Single-share protocol is limited in that if any share gets lost or can not be obtained, the secret message cannot be reconstructed. Typically, having two or more participants with the same share is desired owing to the availability. If the share is replicated, whether some of the replicas are down or unavailable is unimportant.

Among the promising applications of the Internet include high reliability/availability and resource sharing. The reliability/availability improvement is owing to the redundant techniques used on the Internet. Some errors or other unexpected factors of a network may disconnect the communication network, thereby influencing the performance and reliability of the Internet. Several network reliability measures have been defined in addition to the related evaluation methods [12–17]. The evaluation algorithm proposed in Ref. [16] with some modifications can be used to measure the SSRP (Secret Sharing Reconstruct Probability). The computational complexity of the reliability evaluation algorithm is NP-hard [18,19]. Probability generally refers to a system's ability to carry out a requested operation correctly and efficiently. However, allocating shares to appropriate locations for each user over the Internet enhances the secret sharing reconstruct probability. The distribution of shares heavily influences the SSRP. The share assignment problem involves finding a share distribution such that the SSRP measure is maximal. Exhaustive approach may be used to obtain the optimal solution with high computation time. In this paper, we present a simple share assignment (SSA) algorithm based on priority search and some heuristics to achieve better share assignment.

The rest of this paper is organized as follows. Section 2 presents the notation, definitions, and problem statements used herein. Section 3 thoroughly describes the probability model and SSA algorithm to obtain the secret sharing reconstruct probability under the Internet. Section 4 presents some illustrative examples and simulation results to demonstrate the effectiveness of the proposed algorithm. The conclusion of the paper is finally summarized in Section 5.

2. Notation, definitions, and problem statements

Notation

| | |
|----------|--|
| $G(V,E)$ | An undirected graph in which V represents the set of vertices (or nodes) and E represents the edges set of communication links for the network under consideration |
| S_s | the set of shares such that any m of them can be used to reconstruct the secret, S |
| SA_i | set of shares available at node i |

| | |
|------------|---|
| S_i | single-share, or share i in S_s |
| M_i | multiple-share in S_s , such as $M_1 = \{S_1 \cup S_2\}$, $M_2 = \{S_3 \cup S_4 \cup S_5\}$ |
| ME | a subset of E that represents the edges which merged together during the process of finding all MSSTs |
| $p_i(q_i)$ | probability of node, edge, or link i works (fails) |
| $\Pr(E)$ | probability of event E |
| $G - e$ | the graph G with edge e deleted |
| $G + e$ | the graph G with edge $e = (u, v)$ contracted such that node u and v |
| SSA | Simple Share Assignment algorithm |
| SSRP | Secret Sharing Reconstruct Probability |

Definitions

- SST: a share spanning tree that connects the user node (the node that reconstruct the secret under consideration) to some other nodes such that its vertices hold all the required shares.
- MSST: a minimal SST such that there exists no other SST which is a subset of it
- HPF-BF Search: The HPF-breadth first search uses the breadth-first search with high probability first, and keep track of nodes visited. In HPF-BF Search, we start from a node u (the user node) and first visit the node with the highest probability edge incident to u .
- SSRP: The probability that a user can successfully reconstruct a secret over the Internet, i.e., a user can access all the shares required from remote sites despite the faults among the nodes and communication links. The MSSTs connect the user node to other nodes such that these nodes hold all the required shares for the user to reconstruct the secret. The SSRP can be determined by computing the probability that at least one of the MSSTs is operational. Thus the secret sharing reconstruct probability for a user can be defined as the probability that at least one MSST of the user is available. This can be written as

$$\text{SSRP} = \Pr\left(\bigcup^{n_{\text{msst}}} \text{MSST}\right),$$

where n_{msst} denotes the number of MSSTs that reconstruct the secret.

- Shadows (Shares): We can take a secret and divide it into n pieces, called shadows or shares, such that any m of them can be used to recover the secret.
- Multiple-share: a site with multiple-share (or multiple shares) suggests that it contain more than one share.
- Allocation tree: An allocation tree is a tree that was reconstructed from a network and has allocated the share into it.

Problem statements

Informally, the Simple Share Assignment for secret sharing reconstruction is a scheme in which a distinguished host computer, user or client, secret and encrypts a secrets messages, S , and assigns all shares of S to other remote computers. For the share assignment problem, the following information is given:

- (a) network topology,
- (b) the shares required to reconstruct the secret,
- (c) user's location,
- (d) the probability of each node and communication link works.

We also assume that the node and link failures are s -independent.

3. The system model and heuristic algorithm

In this section, we present a probability model of secret reconstruction and algorithms to perform shares assignment under the Internet environment. The Internet can be treated as a computer network that is represented by an undirected graph.

3.1. The system model

Informally, the model is a protocol in which a distinguished processor, or user, selects a secret message and divides it into n pieces, or shares. The divided shares of the secret are then allocated into other processors. A user can access the secret by sending requests to other processors to return the needed shares for reconstruction. Therefore, the model is a probability measure for reconstructing the secret sharing. For example, assume that the commander with anyone of the two vice-commanders is authorized to initiate an order. Herein, we divide a secret message into three shares, S_1 , S_2 , S_3 , and give the commander multiple shares, $\{S_1, S_2\}$ and the vice-commander one each, $\{S_3\}$. Consider the network topology shown in Fig. 1 which consists of six nodes with one multiple shares (node 3) and two replicated single-share (node 4, 5).

A user can either recover the secret or initiate the order by sending a request message to nodes 3, 4, 5 to ask them to return shares S_1 , S_2 , S_3 . In general, the set of nodes and links involved in reconstructing the secret and access its required shares form a tree. A share spanning tree that connects the root node (the user that recreates the secret under consideration) to some other nodes such that its vertices hold all the required shares for reconstructing the secret. Fig. 2 depicts all MSSTs that represent the site which has shares required for a user to reconstruct the secret for the network application in Fig. 1.

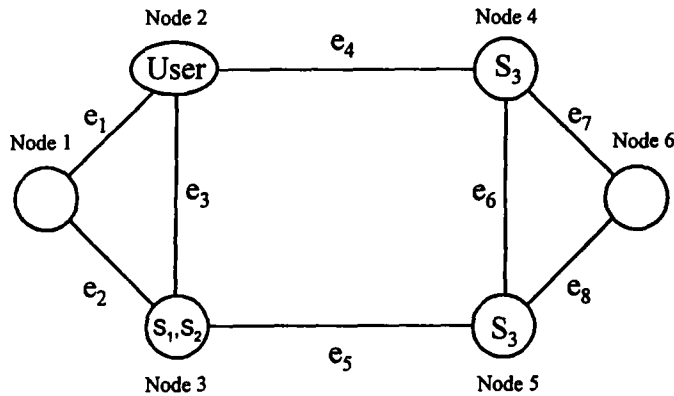


Fig. 1. A simple network.

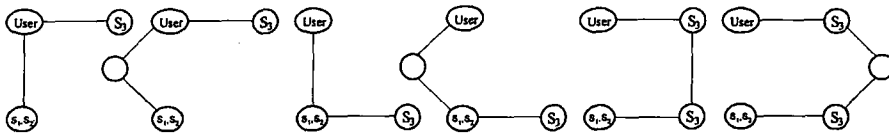


Fig. 2. MSSTs for user to reconstruct the secret.

Let the probability of all nodes and links being operational be 0.9. By doing so, the SSRP of secret S can be computed using conditional probability, as shown below.

$$\begin{aligned}
 \text{SSRP}(S) &= p_{N2} p_{N3} p_{N4} p_{e3} p_{e4} + q_{e3} p_{N1} p_{N2} p_{N3} p_{N4} p_{e1} p_{e2} p_{e4} \\
 &\quad + (1 - p_{e4} p_{N4}) p_{N2} p_{N3} p_{N5} p_{e3} p_{e5} \\
 &\quad + q_{e3} (1 - p_{e4} p_{N4}) p_{N1} p_{N2} p_{N3} p_{N5} p_{e1} p_{e2} p_{e5} \\
 &\quad + q_{e3} (1 - p_{e1} p_{e2} p_{N1}) p_{N2} p_{N3} p_{N4} p_{N5} p_{e4} p_{e5} p_{e6} \\
 &\quad + q_{e3} q_{e6} (1 - p_{e1} p_{e2} p_{N1}) p_{N2} p_{N3} p_{N4} p_{N5} p_{N6} p_{e4} p_{e5} p_{e7} p_{e8} \\
 &= 0.59049 + 0.0478297 + 0.1121931 + 0.00908764 + 0.0129618 \\
 &\quad + 0.0010499 \\
 &= 0.7736121,
 \end{aligned}$$

where p_{Ni} denotes the probability of node i working and p_{ei} represents the probability of edge i working, $q_{Ni} = (1 - p_{Ni})$, and $q_{ei} = (1 - p_{ei})$.

The probability of reconstructing secret sharing on the Internet can be evaluated by the following two steps: finding all the MSSTs for that user and computing the probability of the MSST to evaluate the SSRP. All disjoint MSSTs can be generated by the following `minimal_share_spanning_tree` algorithm with some modifications from the reliability algorithm in Ref. [16].

The MSST algorithm employs the graph expanding procedure and guarantees no replicated share spanning trees generated during the expansion of computation tree. The MSST algorithm executes several reduction methods [20] to reduce the size of the problem and to simplify the probability evaluation. The complete MSST algorithm is given below.

Algorithm Minimal_Share_Spanning_Tree (G, u);

Input: $G = (V, E, S_s)$ (the original network graph), and u (the user node)

Output: the SSRP, Secret Sharing Reconstruct Probability

begin

```

repeat                                     /* reduce the original network graph  $G$  */
  perform degree-1 reduction                 /* removes degree-1 nodes which
  contain no
    required shares and removes their incident edges */
  perform parallel reduction                 /* Suppose  $e_a = (u, v)$  and  $e_b = (u, v)$ 
  are two
    parallel edges in  $G$ . We can reduce these redundant edges into a
    single
    edge  $e_c = (u, v)$  such that  $p_{ec} := (1 - q_{ea} \times q_{eb})$  */
  perform series reduction                   /* Let  $e_a = (u, v)$  and  $e_b = (v, w)$  be two
  series
    edges in  $G$  such that  $\text{degree}(v) = 2$  and node  $v$  contains no required
    shares. We can remove node  $v$  and replace edges  $e_a$  and  $e_b$  by a
    single
    edge  $e_c = (u, w)$  such that  $p_{ec} := (p_{ea} \times p_v \times p_{eb})$  */
  perform degree-2 reduction                 /* Suppose node  $v$ , with node
  degree = 2, is a
    reducible node. Then we can apply series reduction on node  $v$  and
    move the shares within node  $v$  to one of its adjacent nodes. */
until no reductions can be made
let  $G'$  be the network graph after reduction step
 $FOUND := \emptyset$ 
 $ME := \emptyset$ 
 $FIND\_SST(G', ME)$                        /* call  $FIND\_SST$  to find SSTs */
for all  $s, t \in FOUND$  do                 /* remove the SSTs which are not MSSTs */
  if  $(t \cap s) = s$  then remove from  $FOUND$ 
  else if  $(t \cap s) = t$  then remove  $S$  from  $FOUND$  endif
endif
repeat
  nodes are introduced by the intermediary of edged incident to them apply
  the reliability algorithm to all MSSTs in  $FOUND$  to evaluate SSRP
output the SSRP
end MSST

```

Function *Find_SST*(*G*, *ME*)

```

begin
  if there are no SSTs in G then return(0) endif
  if there exists one node v such that  $SA_v \supseteq S_s$  then FOUND := FOUND  $\cup$ 
  {ME}
  return endif
  for all  $e_i \in$  {the set of edges incident to the node that contains the user}
  do
    FIND_SST(G +  $e_i$ , ME  $\cup$  { $e_i$ })
    G := G -  $e_i$ 
    remove the irrelevant components from G
    if there is no SSTs in G then return endif
  repeat
end FIND_SST

```

The previous example in Fig. 1 is used in the following to illustrate the MSST algorithm. Fig. 3 depicts the process of finding all MSSTs of MSST algorithm. The list of all MSSTs generated in *FOUND* are $\{(e_4, e_9), (e_4, e_5, e_{10}), (e_5, e_9)\}$. The nodes are then introduced by the intermediary of edges incident to them. The set of all MSSTs of user to reconstruct the secret is $\{(e_4, e_9, N2, N3, N4), (e_4, e_5, e_{10}, N2, N3, N4, N5), (e_5, e_9, N2, N3, N5)\}$. Assume that the probability of all nodes and links working is 0.9. Applying the terminal reliability algorithm, e.g. [14,16], to the above MSSTs allows us to compute the SSRP of secret *S*, as shown below.

$$\begin{aligned}
 \text{SSRP}(S) &= \Pr\left(\bigcup^{\text{msst}} \text{MSST}\right) \\
 &= p_{N2} p_{N3} p_{N4} p_{e4} p_{e9} + q_{e9} p_{N2} p_{N3} p_{N4} p_{N5} p_{e4} p_{e5} p_{e10} \\
 &\quad + (1 - p_{e4} p_{N4}) p_{N2} p_{N3} p_{N5} p_{e5} p_{e9} \\
 &= 0.6383197 + 0.0140117 + 0.1212807 = 0.7736121.
 \end{aligned}$$

3.2. A simple example of SSRP analysis

Example 1. Herein, a secret is divided into three shares: S_1 , S_2 , and S_3 . Assume that one participant is more important than the others and give that participant multiple shares, $\{S_1, S_2\}$. We allocate the shares into different sites over the Internet. To recreate the secret, the user must obtain the three shares together. Consider the network topology shown in Figs. 4 and 5 which consists of six nodes that allocated one multiple-share and three different single-share in redundant manner. Let the probabilities of all nodes and links being

operational be 0.9. Fig. 4 depicts an optimal share assignment to this network topology and Fig. 5 displays a poor one. These allocations differ only in the share assignments, while the other parameters are identical.

This example demonstrates how share assignments affect the secret sharing reconstruct probability. Thus, the means of share assignment significantly affects the probability of secret sharing reconstruction. The following section presents a simple share assignment algorithm such that the probability of secret sharing reconstruction can be maximized.

3.3. Preliminary strategy of the simple share assignment

3.3.1. Underlying concept

The concept of full set is introduced as follows. Full set is defined as a set containing all the required shares to reconstruct the secret under the Internet. For an (m, n) - threshold secret sharing scheme, the full set of secret S is the set of all shares $\{S_1, S_2, \dots, S_n\}$ which is a unique case of $m = n$. Full set is the basic requirement of the SSRP. Recall the definition of the minimal share spanning tree. The MSST for the secret must contain all the elements in the full set. This paper also concerns itself with the combination of the multiple-shares for the participants. By doing so, we attempt to obtain a larger number of groups with the full set. Thus, the proposed approach is carried out to obtain an acceptable SSRP value. To prove this theorem, recall how SSRP is computed. SSRP is the probability that at least one of the MSSTs is working. The MSST spans one of the full sets we just discovered. If more full sets are in the network, the number

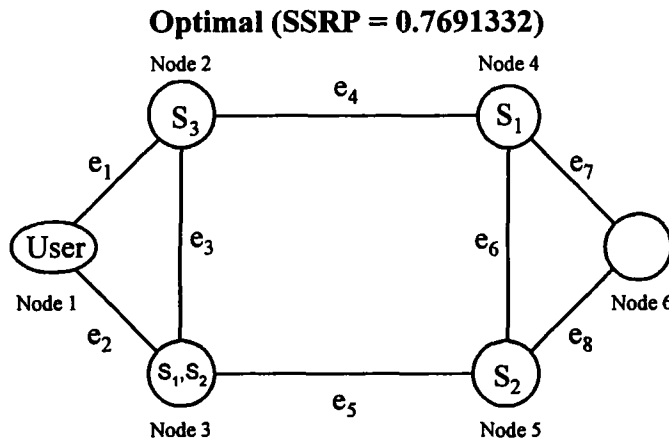


Fig. 4. The optimal share assignment of Example 1.

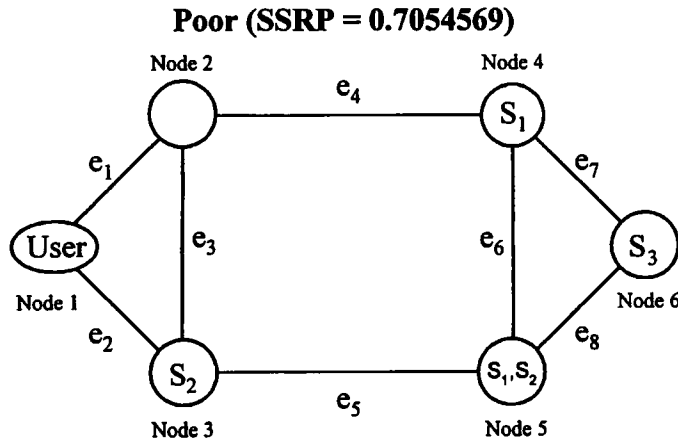


Fig. 5. The poor share assignment of Example 1.

of MSSTs is also larger. Hence, SSRP should be greater owing to $P(A \cup B) \geq P(A)$ by the set theory.

Closely study the above example reveals the following:

- (a) The transfer probability for a share is data link dependent. The fewer the data links capable of transferring the required data implies a higher transfer probability;
- (b) The MSST with less nodes and links is more reliable; and
- (c) The SSRP is the probability of the union of the number of MSSTs and can be enhanced by grouping the required shares as close to the user node as possible.

Based on above observations, we propose a simple heuristic approach for share assignment. Underlying concepts of the heuristic approach are as follows:

1. Allocate the more influential shares first. (Share M_1 is more influential than share M_2 if and only if the number of shares of M_1 exceeds that of M_2);
2. Complete the full set which occupies a minimum number of nodes;
3. Allocate the shares to the more reliable sites; and
4. Allocate all shares as close to the user's location as deemed feasible.

The above strategies are applied to develop our share assignment algorithm. The SSA algorithm is based on priority search and some heuristics, and is summarized in the following two parts.

Part 1: Classify and determine the allocation order of the shares.

Part 2: Use the HPF_BF algorithm to reconstruct the network into a tree topology, then sequentially allocate these shares into the reconstructed tree according to the order obtained in Part 1.

3.3.2. Reconstructing network into tree topology

Next, the network is reconstructed into a tree topology by using the HPF-BF search algorithm to travel all the edges. In HPF-BF search, we start from a node u (the user node) and visit the node from the highest to the lowest probability edge incident to u . Then all nodes adjacent to those nodes are visited, and so on. The HPF-BF algorithm uses the breadth-first search with high probability first to construct a HPF-BF tree by including only edges that lead to newly visited nodes. All those edges cumulatively form a tree with user node as its root.

The complete HPF-BF search algorithm is given below.

Algorithm HPF-BF Search (G, u);

Input: $G=(V, E)$ (an undirected connected graph), and u (the user node).

Output: $T=(V, E')$: a tree T with node set V and edge set E' .

begin

$S := \{u\}$

$L := \{e_i \mid \forall e_i \in E \text{ and } \exists v_i \in S, \text{ such that } e_i \text{ is incident to } v_i\}$

$E := E - L$

$V := V - S$

$E' := \emptyset$

$S := \emptyset$

repeat

repeat

$e := \{e_i \mid \text{select an } e_i \in L \text{ which has the maximum } p_{e_i}\}$

$L := L - \{e\}$

if $E' + \{e\}$ **does not have cycle** **then** $E' := E' + \{e\}$ **endif**

$S := S + \{v_i \mid \forall v_i \notin S \text{ and } e \text{ is incident to } v_i\}$

until $L = \emptyset$

$L := \{e_i \mid \forall e_i \in E, \forall e_i \notin E', \text{ and } \exists v_i \in S, \text{ such that } e_i \text{ is incident to } v_i\}$

$V := V - S$

$S := \emptyset$

until $V = \emptyset$

end HPF-BF

3.3.3. C. An example of HPF-BF algorithm

Fig. 6 depicts a simple example of using HPF-BF algorithm to construct the allocation tree. The parenthesized numbers associated with the nodes denote the order in which the nodes were searched by HPF-BF search algorithm.

3.4. The complete SSA algorithm

The SSA algorithm consists of two steps. First, the algorithm determines the allocation order of the shares. Next, the algorithm executes the HPF-BF search

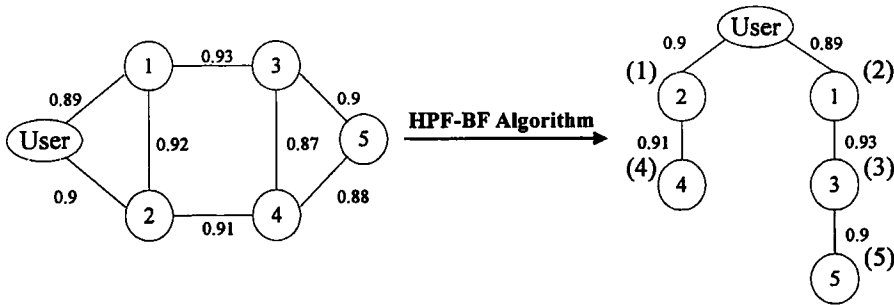


Fig. 6. A example of the HPF-BF Search algorithm.

and sequentially allocates the shares into the tree. The allocation order of the shares is determined by using the following procedures.

- (a) Calculate the number of shares of each multiple-share.
- (b) Allocate the multiple-share with more unallocated shares first.
- (c) Calculate the number of unallocated shares of each multiple-share.
- (d) Select and allocate the share which contains more unallocated shares to complete the full set. Select the multiple-share that contains more shares first to break the tie of the number of unallocated shares.
- (e) If all shares have been set to “allocated” then set all shares to “unallocated”. A share assignment, which has the greatest number of full set groups, is our preferred choice.
- (f) Repeat steps c, d, and e until all the shares have been allocated.

Algorithm Simple_Share_Assignment;

Input: $G = (V, E)$ (an undirected connected graph), u (the user node), and M (the set of shares to be allocated).

Output: an allocation tree.

begin

$S := \{v \mid v \text{ is the node that contains the user}\}$

$M' := \text{IN_ORDER_OF}(M)$ /* M is the set of single and multiple shares */

$V := \text{HPF-BF}(S)$

repeat

$v := \text{FIRST_NODE}(V)$

$V := V - \{v\}$

$m := \text{FIRST_SHARE}(M')$

$M' := M' - \{m\}$

allocate multiple-share (or share) m into node v

until $M' = \emptyset$

end SSA

Function *IN_ORDER_OF*(M)

```

begin
 $M' := \emptyset$ 
  repeat
    for  $\forall m \in M$  do
      count the number of unallocated share in multiple-share (or share)  $m$ 
       $m := \{m' \mid m' \text{ is in set } M \text{ with the largest number of unallocated share}\}$ 
       $M' := M' + \{m\}$ 
       $M := M - \{m\}$ 
    if all shares have been set to “allocated” then
      set all shares to “unallocated” endif
  until  $M = \emptyset$ 
  return( $M'$ )
end IN_ORDER_OF

```

Function *FIRST_SHARE*(M');

```

begin
  return the first multiple-share (or share) in an order set  $M'$ 
end FIRST_SHARE

```

Function *FIRST_NODE*(V)

```

begin
  return the first node in an order set  $V$ 
end FIRST_NODE

```

3.5. Time complexity analysis

The computational complexity of the SSA algorithm is the sum of the run time of determining the allocation order of shares and finding the next location for share to be allocated. First, arranging the allocation order of shares take $O((k+n)n)$ comparisons in the worst case. Second, the HPF-BF search algorithm takes $O((e+v)\log v)$ to construct the HPF-BF tree in the worst case. Therefore, the time complexity of algorithm SSA is $O((k+n)n) + (e+v)\log v$, where v denotes the number of nodes (vertices), e denotes the number of edges, n denotes the number of single-share, and k denotes the number of multiple-share.

4. Illustrative examples and simulation results

In this Section, some examples and numerical results are employed to illustrate the effectiveness of our approach.

4.1. An eight-node communication network

Consider the network topology in Fig. 7 which consists of eight nodes and eleven communication links. Node 2 contains the user and the other nodes are regarded as share-storage locations. Assume that the probabilities of all nodes and links being operational are set to 0.9.

The secret sharing reconstruct probability of the secret S is evaluated by applying the algorithm presented in Section 3.1. In this example, the secret is divided into four shares. Assume that four single-share and two multiple-share are to be allocated. The multiple-share may contain more than one kind of single-share. They are listed as follows.

The set of shares of secret S : $S_s = \{S_1, S_2, S_3, S_4\}$

Four single-share: S_1, S_2, S_3, S_4

Two multiple-share: $M_1 = \{S_1, S_3\}, M_2 = \{S_2, S_4\}$

To reconstruct the secret message, the user must access all four divided shares ($m=n$) together. Now we use HPF-BF search, starting from the user node, the root of the tree, to construct the allocation tree. By doing so, we obtain the allocation order of shares which is $M_1, M_2, S_1, S_2, S_3, S_4$. A tree that is constructed from a network and allocated shares into it is called an allocation tree. Fig. 8 depicts an allocation tree and the SSRP value for this communication network.

Results obtained from our algorithm are compared to those of a Random Allocation algorithm. We computed the SSRP of all random distributions. Table 1 summarizes the maximum and minimum SSRP of simulation results based on 5000 samples.

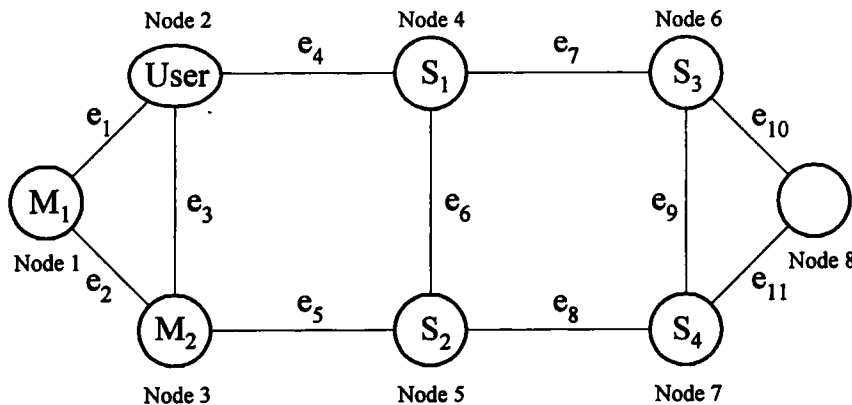


Fig. 7. An eight-node communication network.

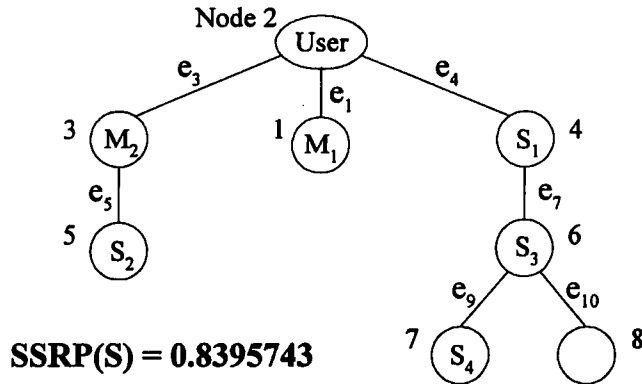


Fig. 8. Allocation tree for the eight-node network.

4.2. The pacific basin network

Fig. 9 displays the Pacific Basin network topology, which consists of nineteen sites and twenty-five communication links. Site 7 contains the user and the other sites are considered share-storage locations. Let the probabilities of all nodes and links being operational for the Pacific Basin network be 0.9.

In this example, we divide the secret into seven shares. Assume that seven single-share and four multiple-share are to be allocated, and at least three qualified multiple-share participants can recover the secret. They are listed below.

The set of shares of secret S : $S_s = \{S_1, S_2, S_3, S_4, S_5, S_6, S_7\}$

Seven single-share: $S_1, S_2, S_3, S_4, S_5, S_6, S_7$

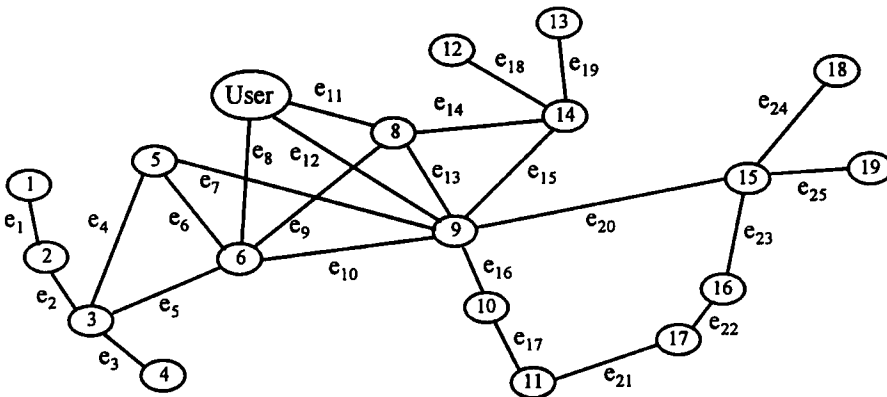
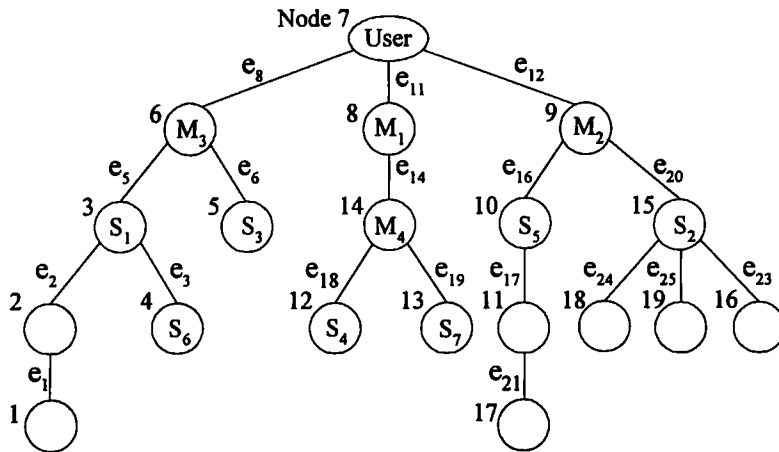


Fig. 9. Pacific Basin network.



SSRP(S) = 0.8558829

Fig. 10. Allocation tree for the Pacific Basin network.

Four multiple-share: $M_1 = \{S_2, S_4, S_6, S_7\}$, $M_2 = \{S_1, S_2, S_3\}$, $M_3 = \{S_3, S_4, S_5, S_6\}$, $M_4 = \{S_1, S_5, S_7\}$

The user must access all seven divided shares ($m = n$) together to recover the secret. We first use HPF-BF search, starting from the user node, to reconstruct the network into a tree topology. Then we obtain the allocation order of shares which is $M_1, M_3, M_2, M_4, S_1, S_3, S_5, S_2, S_7, S_4, S_6$. Fig. 10 depicts the allocation tree for the Pacific Basin network.

Results obtained from our algorithm are compared to those of a Random Allocation algorithm. We computed the SSRP of all random distributions. Table 2 summarizes the maximum and minimum SSRP of simulation results based on 5000 samples.

Tables 1 and 2 indicate that the proposed algorithm can obtain near optimal solutions. Result obtained from our algorithm in the Pacific Basin network indicates a slightly less than the optimal due to the use of the HPF-BF search algorithm. The HPF-BF algorithm can not guarantee that we will obtain the maximal number of MSSTs. Exhaustive approach may be used to obtain the optimal solution with very expensive price. Previous investigations have demonstrated that exhaustive approach is an NP-hard problem [19,21].

Table 1
Simulation results of eight-node network

| Network topology | SSA algorithm | Random algorithm |
|--------------------|---------------------|--------------------------------------|
| Eight-node network | SSRP(S) = 0.8395743 | Max. = 0.8395743 Min. = 0.5464987 |

Table 2
Simulation results of Pacific Basic network

| Network topology | SSA algorithm | Random algorithm |
|-----------------------|-------------------------|--------------------------------------|
| Pacific Basic network | SSRP(S) = 0.8558829 | Max. = 0.8669108 Min. = 0.3054878 |

5. Conclusion

Rapid expansion of the Internet is fueled by its ability to promote information sharing and to offer high availability. The Internet today is a widespread information infrastructure, reaching not only the technical fields of computer communications but throughout society. As the Internet has evolved, one of the major tasks is how to answer the request efficiently and reliably. Information on the Internet can be stored close to its normal point of use, thereby reducing both response times and communication costs. Effectively distributing shares to appropriate sites is the basic consideration for share assignment problem. In this paper, we present a probability model for secret sharing reconstruction and an algorithm for shares assignment and deal with two types of shares. Illustrative examples demonstrate the underlying concept of the model and the feasibility of our approach. According to simulation results, the proposed algorithm obtains approximate solutions efficiently. Particularly the (m, n) -threshold scheme is realized and evaluated based on the probability of secret sharing reconstruction.

References

- [1] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd Ed., Wiley, New York, 1996.
- [2] R.L. Rivest, A. Shamir, L.M. Adleman, A method for obtaining digital signatures and public-key cryptosystems, *Communication of the ACM* 21 (2) (1978) 120–126.
- [3] W.P. Lu, M.K. Sundareshan, Enhanced protocols for hierarchical encryption key management for secure communication in internet environments, *IEEE Transaction on Communications* 40 (1992) 658–670.
- [4] T. Hwang, W.C. Ku, Repairable key distribution protocols for internet environments, *IEEE Transaction on Communications* 43 (1995) 1947–1949.
- [5] M. Ito, A. Saito, T. Nishizeki, Secret Sharing Scheme Realizing General Access Structure, *Proc. IEEE Globecom'87*, Tokyo, 1987, pp. 99–102.
- [6] C.S. Lai, L. Harn, J.Y. Lee, T. Hwang, Dynamic threshold scheme based on the definition of cross-product in an N -dimensional linear space, *Journal of Information Science and Engineering* 7 (1991) 13–23.
- [7] H.M. Sun, S.P. Shieh, On dynamic threshold schemes, *Information Processing Letters* 52 (1994) 201–206.
- [8] A. Shamir, How to share a secret, *Communication of the ACM* 22 (1979) 612–613.

- [9] G.R. Blakley, Safeguarding cryptographic keys, Proceedings of the AFIPS 1979 National Computer Conference, vol. 48, New York, 1979, pp. 313–317.
- [10] D.R. Stinson, An Explication of Secret Sharing Schemes Design, Codes and Cryptography, vol. 2, 1992, pp. 357–390.
- [11] G.J. Simmons, An Introduction to Shared Secret and/or Shared Control Schemes and their Application, Contemporary Cryptology, IEEE Press, New York, 1991, pp. 441–497.
- [12] K.K. Aggarwal, S. Rai, Reliability evaluation in computer communication networks, *IEEE Transaction on Reliability R-30* (1981) 32–35.
- [13] W.R. Kevin, Factoring algorithms for computing K-terminal network reliability, *IEEE Transaction on Reliability R-35* (1986) 269–278.
- [14] S. Hariri, C.S. Raghavendra, SYREL: A symbolic reliability algorithm based on path and cutset methods, *IEEE Transaction Computers* 36 (1987) 1224–1232.
- [15] C.S. Raghavendra, V.K.P. Kumar, S. Hariri, Reliability analysis in distributed systems, *IEEE Transaction Computers* 37 (1988) 352–358.
- [16] M.S. Lin, D.J. Chen, New reliability evaluation algorithms for distributed computing systems, *Journal of Information Science and Engineering* 8 (1992) 353–391.
- [17] D.J. Chen, T.H. Huang, Reliability analysis of distributed systems based on a fast reliability algorithm, *IEEE Transaction on Parallel and Distributed Systems* 3 (2) (1992) 139–154.
- [18] A. Rosenthal, A computer scientist looks at reliability computations, in: *Reliability and Fault tree Analysis SLAM*, 1975, pp. 133–152.
- [19] L.G. Valiant, The complexity of enumeration and reliability problems, *SIAM Journal of Computing* 8 (1979) 410–421.
- [20] O.R. Theologou, J.G. Carlier, Factoring and reductions for network with imperfect vertices, *IEEE Transaction on Reliability* 40 (1991) 210–217.
- [21] M.O. Ball, Computational complexity of network reliability analysis: an overview, *IEEE Transaction on Reliability R-35* (1986) 230–239.