Study of a Secure Backup Network Mechanism for Disaster Recovery and Practical Network Applications

Noriharu Miyaho, Yoichiro Ueno, Shuichi Suzuki,
Kenji Mori
Department of Information Environment,
Tokyo Denki University,
Inzai-shi, Chiba, 270-1382 Japan
e-mail: miyaho@sie.dendai.ac.jp,
ueno@sie.dendai.ac.jp, ssuzuki@sie.dendai.ac.jp,
mori@ine.sie.dendai.ac.jp

Kazuo Ichihara
Net&Logic Inc.,
Setagaya-ward, Tokyo, Japan
e-mail: Ichihara@nogic.net

Abstract—**A practical mechanism for a file-backup system concept is proposed. In this paper, it is demonstrated that a highly secure file backup mechanism can be achieved by combining technologies that implement the following: spatial random scrambling of file data, subsequent random fragmentation of the file, the duplication and encryption of each file fragment using a stream cipher code in each encryption stage, and the corresponding notification of the history data of the encryption key code sequence used in "encryption metadata". If a disaster should occur in the data backup center, prompt data recovery can be easily and securely achieved by making use of a massive number of widely distributed wired PCs, mobile PCs, cellular phones managed by multiple supervisory servers which are secretly diversified but functionally combined. In an experimental evaluation, encryption performance, spatial scrambling performance and the average response time from the Web server have been estimated in terms of the memory load characteristics of the data center. Discussion is also provided on an effective shuffling algorithm to determine the dispersed location sites. Finally this paper describes a system configuration for a practical application, which will be soon commercialized.**

Keywords-disaster recovery; backup; metadata; distributed processing; shuffle algorithm

I. Introduction

Innovative network technology to guarantee, as far as possible, the security of users' or institutes' massive files of important data from any risks such as an unexpected natural disaster, a cyber-terrorism attack, etc., is becoming more indispensable day by day. To meet this need, technology is required that can be realized with affordable maintenance operation costs and that provides high security.

For this application, Data Grid technology is expected to provide an effective and economical back up system by making use of a very large number of PCs whose resources are not fully utilized. In particular, Data GRID technology using a distributed file data back-up mechanism will be utilized by government and municipal offices, hospitals, insurance companies, etc., to guard against the occurrence of unexpected disasters such as earthquakes, big fires and storms.

However, these methods involve high operation costs, and there are a lot of technical issues to be solved, in particular relating to security and prompt restoration in the event of disasters occurring in multiple geographical locations.

The background leading to the need for the proposed economical backup system and its objectives are shown in Figure 1.

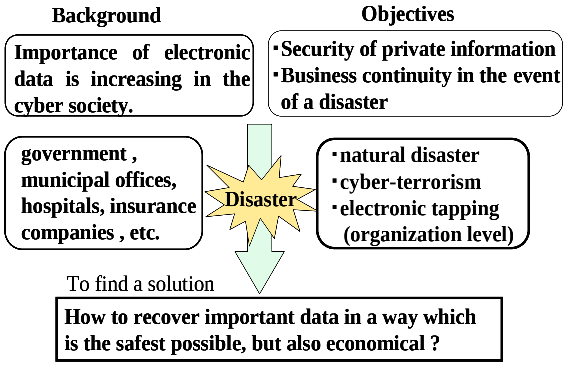
Background
Objectives
Importance of electronic data is increasing in the cyber society.
·Security of private information
·Business continuity in the event of a disaster
government , municipal offices, hospitals, insurance companies , etc.
Disaster
·natural disaster
·cyber-terrorism
·electronic tapping (organization level)
To find a solution
How to recover important data in a way which is the safest possible, but also economical ?

Figure 1. Background and objectives of the proposed system

On the other hand, there is a network infrastructure which can be used to distribute and back-up a great number of data files and a large number of remote office personal computers, cellular phones, and smart phones such as iPhone4 can be utilized for this purpose.

In this paper we propose an practical file back-up concept which makes use of an effective ultra-widely distributed data transfer mechanism and a high-speed encryption technology, based on the assumption that we can use a small portion of the memory of a large number of PCs and cellular phones that are in use in daily life, to efficiently realize safe data backup with an affordable maintenance operation cost [1].

Figure 2 shows a comparison of the proposed method with the conventional method in terms of network utilization. The principal differences of the proposed system are as follows. (1) It does not require the use of expensive leased lines. (2) It only utilizes otherwise unused network resources such as unused network bandwidth, unused memory capacity of PCs, cellular phones and smart phones, etc. (3) It can cipher a number of important data files at the same time using spatial scrambling and random dispatching technology. (4) As the number of user companies increases, the security against being deciphered illegally increases accordingly. (5) The maintenance cost can be drastically reduced. In addition, since it adopts a stream cipher, the speed of encryption of data increases, so it can also be applied to secure streaming video transmission services.

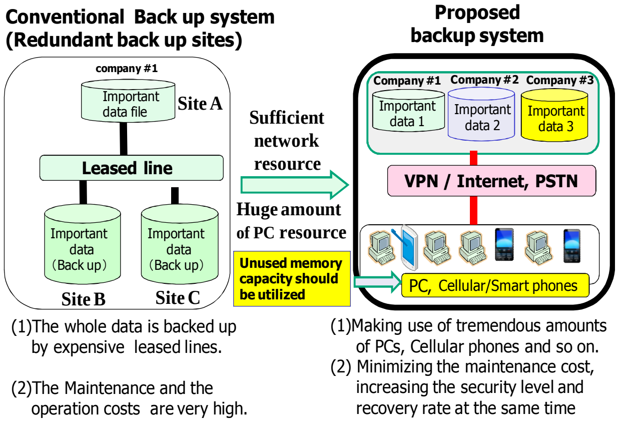
Conventional Back up system
(Redundant back up sites)
company #1
Important data file
Site A
Leased line
Important data (Back up)
Important data (Back up)
Site B
Site C
Sufficient network resource
Huge amount of PC resource
Unused memory capacity should be utilized
Proposed backup system
Company #1
Company #2
Company #3
Important data 1
Important data 2
Important data 3
VPN / Internet, PSTN
PC, Cellular/Smart phones
(1)The whole data is backed up by expensive leased lines.
(2)The Maintenance and the operation costs are very high.
(1)Making use of tremendous amounts of PCs, Cellular phones and so on.
(2) Minimizing the maintenance cost, increasing the security level and recovery rate at the same time

Figure 2. Comparison of the proposed system with a conventional data backup system

In general, encryption technology can use two types of technology, that is, block cipher technology or stream cipher technology.

In case of the block cipher technology, the data is divided into successive blocks and each block is processed separately for point-to-point systems; as a result, the encryption speed becomes quite slow. As the data volume increases, the required processor and memory cost increases in an exponential manner.

On the other hand, in case of the stream cipher, the input data is simply operated on bit by bit using a simple arithmetic operation. Therefore, high-speed processing becomes feasible. These are the fundamental differences between the two cipher technologies [2].

When an ultra-widely distributed file data transfer technology, a file fragmentation technology and an encryption technology are used together, then, quite different characteristics from a point of cipher strength arise. It is possible to combine the use of technologies such as the spatial random scrambling of all data files, the random fragmentation of the data files, the corresponding encryption and duplication of each file fragment using a stream cipher, and, in addition, the corresponding notification of the history data including the encryption key code sequence, which we call "encryption metadata". When these are all combined is it possible to realize an effective, economical and prompt data backup system [3][4].

In this case, PC terminals deployed in remote offices or cellular phones can be used by making effective use of combined stream cipher and distributed data transfer technologies. By making use of the proposed combined technology, illegal data recovery by third party interception becomes almost impossible and an extremely safe data backup system can be realized at reasonable cost.

The proposed technology can increase both of a cipher code strength and a data recovery rate. The economical operation cost achieved by using the proposed backup technology, while high-speed network technology also makes this achievement more practicable.

To realize the proposed disaster recovery mechanism, the following three principal network components are required: (1) a secure data center, (2) several secure supervisory servers, and (3) a number of client nodes such as PCs or cellular phones.

We clarify the relationships between data file capacity, number of file fragments and number of duplications in case of the disaster recovery hereafter [5]-[11].

In this paper we briefly describe related work in Section II, and discuss the basic configuration of the proposed system architecture in Section III, and the basic characteristics of the proposed system in Section IV. The uniformity of the distribution of file fragments is discussed in Section V, the secure and secret decentralization of the proposed system in Section VI, the encryption and spatial scrambling performance in Section VII, user friendly service level assurance in Section VIII, and an example of a substantial system in Section IX. Finally, we provide our conclusions from these studies in Section X.

II. RELATED WORK

Conventionally, leased lines have been adopted as the method of connection between large-scale data centers and the data GRID recovery center for a large-scale file backup [12]. In most cases, all the technologies used are based on duplication, such as duplication of a data center, duplication of an access line or duplication of a transit line, etc.

However, considering that an earthquake may cause fiber cable failures over a wide area and shut down several communication links, and also destroy the backup data, redundant BGP (Border Gateway Protocol) routing needs to be improved in order to divert traffic along redundant backup paths by making use of effective collaboration among different operators [13].

At present, it is not easy to apply such pre-arranged collaboration when an unexpected disaster occurs. Although a reliable peer to peer communication system using Grid computing environments has been proposed, and effective scheduling algorithms have also been proposed, the required efficient and safe encryption mechanisms have not yet been developed [14]-[17]. Previously, the concept of file data fragmentation and dissemination technologies have been investigated [18]-[20]. Other related studies include the concept of a distributed file backup system [21][22].

However, in these studies, the secure and fast uniform distribution of a fragmented data file after the spatial scrambling and the subsequent encryption technology by effectively making use of a series of time dependent metadata have not yet been sufficiently clarified.

Furthermore, the service providers could not make use of the meta-data containing the history of encryption key code sequences for the secure file back up, so far. In addition, a user friendly data backup concept related to the required security strength level and recovery rate parameters has not been taken into consideration at all.

III. Basic Configuration of the Proposed System Architecture and Its Functions

The proposed file backup mechanism consists mainly of three network components and two VPNs as shown in Figure 3. The main functions of the proposed network components which are Data Center, Client Nodes and Supervisory center, are described as follows.

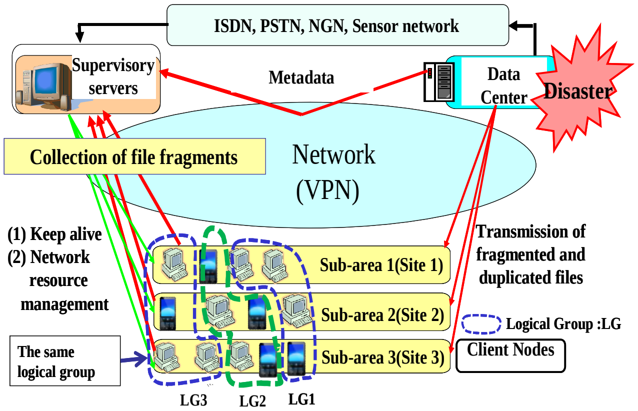

Figure 3. Basic network configuration and interfaces

The client nodes are connected to a data center and to a Supervisory center by virtual private networks (VPNs) and classified into three logical groups, such that copies of the same file fragment files are located redundantly in different geographical areas.

The Supervisory service center acquires the history data composed of the encryption key code sequence (metadata) from the Data Center via the VPN. The precise functions and the corresponding data back-up procedures are described in more detail in section IV. As shown in Figure 3, the basic procedure is as follows.

(1) The Data center sends the fragmented file to some of the sub-areas.
(2) The Data center sends the metadata used for deciphering the series of fragments. Metadata are composed of encryption key sequences, one for each file fragment.
(3) The Supervisory servers collect the required fragments from some of the sub-areas.
(4) After all the required data has been collected, decryption is executed.

Figure 4 shows the proposed system in more detail. The functions of the main components are as follows.

A. *Data Center*

The main task of data center is the encryption and distribution of important file data. When the data center receives file data, it processes it with the following sequence.

1)1ST ENCRYPTION

The data center encrypts that whole file using a stream cipher code. In addition, the data center records the key of 1st encryption as "encryption metadata".

2) Spatial random scrambling

After 1st encryption, the data center carries out spatial random scrambling. Spatial random scrambling spreads data words spatially across the whole file.

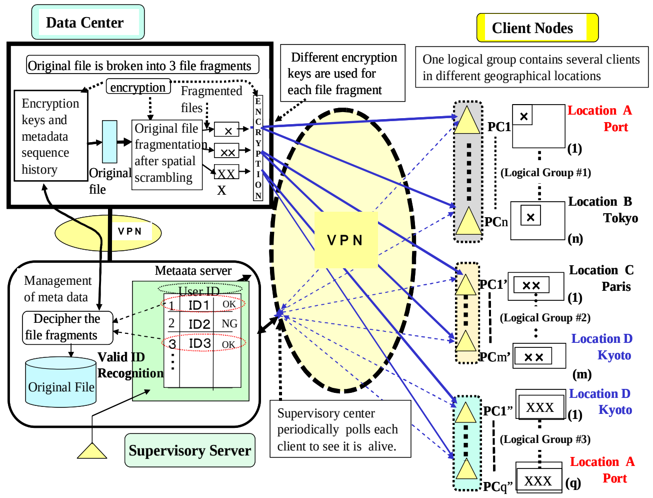

Figure 4. Basic configuration of the proposed system

3)Fragmentation

After spatial random scrambling, the data center divides the file data into a number of fragments. The data center records the size and offset of all fragments as "encryption metadata".

4)Duplication

Next, the data center makes a number of copies of all fragments.

5)2nd encryption

The data center encrypts all the duplicated fragments with different keys.

So, if there are X fragments, and Y copies are made of each fragment, the number of 2nd encryption keys is X×Y.

The data center records all the 2nd encryption keys as "encryption metadata"

6) Distribution

The data center distributes all the duplicated and encrypted fragments to client nodes in random order. The data center records all relations between duplicated fragments and client nodes as "encryption metadata".

7) Backup encryption metadata

Finally, the data center sends the "encryption metadata" to supervisory servers, and can then erase all data including original the file data, temporary data and encryption metadata. So, after this process, no one can steal or recover the original file data from the data center.

The original data can be deleted at the data center. It implies that the data center does not hold a copy of the original data at all. In this case, the fragmented data would be the only copy of the data, and would need to be used every time anyone wanted to access the data, not just in the event of a data center being damaged in a disaster.

B. *Client nodes*

Client nodes have four tasks: (1) receiving fragments of fragmented files, (2) deletion of old file fragments, (3) sending keep alive messages, and (4) delivery of stored fragments for transmission.

1) Receiving a file fragment

Client nodes receive duplicated fragments from data center, and store them in their exclusive storage area for this mechanism.

2) Deletion of old file fragments

As a client node's storage area is limited, client nodes have to erase old duplicated fragments.

3) Notifying presence to keep alive

Client nodes periodically send "keep alive" packets to the supervisory servers.

4) Transmission for delivery

If a client node receives a transmission request from a supervisory server and authenticates it successfully, it transmits the specified duplicated fragments to the supervisory server that issued the request.

Generally, the selection of client nodes will be determined by the contract with the service providers or selected among the public organizations or the specific corporation.

C. *Supervisory servers*

Supervisory servers have three tasks, (1) receiving "encryption metadata", (2) receiving keep alive packets, and (3) recovering of the original file.

1) Receiving "encryption metadata"

The data center sends "encryption metadata" to the supervisory servers at the end of the back-up process. The supervisory servers receive such "encryption metadata" and store it in their databases.

2) Receiving "keep alive" packets

The supervisory servers receive "keep alive" packets from client nodes, and store information about which nodes are alive in their databases. The supervisory servers send this information about working nodes to the data center to prevent it from trying to connect to dead client nodes. Moreover, the supervisory servers use this information to select appropriate client nodes at the time of data recovery.

3) Recovering of original file

If a user needs to recover back-up file data, he sends a request for file recovery to the supervisory servers. Every requests are sent to all supervisory servers. The supervisory servers send transmission requests to appropriate client nodes, according to the information on which nodes are alive. When the supervisory servers have received all the requested fragments, they start to decrypt the data using the "encryption metadata" in their database. When the data center should back up a series of a user's updated data files, which are composed of several generation files for example, the supervisory servers are applied to recover the distributed files of the corresponding generation even when the disaster has not happened in order to constantly ascertain the corresponding file's safety.

In the backup mechanism, the three network components are connected to each other via VPNs. The VPN between the data center and client nodes, and that between client nodes and supervisory servers can use the Internet. However, the VPN between the data center and the supervisory servers should use a leased line for security reasons.

IV. BASIC CHARACTERISTICS OF THE PROPOSED SYSTEM

A. *Overview of the required functions*

This section discusses the characteristic functions in the proposed backup mechanism as follows.

1) Spatial random scrambling

An example of an algorithm of spatial random scrambling is shown in Fig. 5. The operator "⊕" indicated in Figure 5 means a reversible operation, such as exclusive-or, addition, or subtraction.

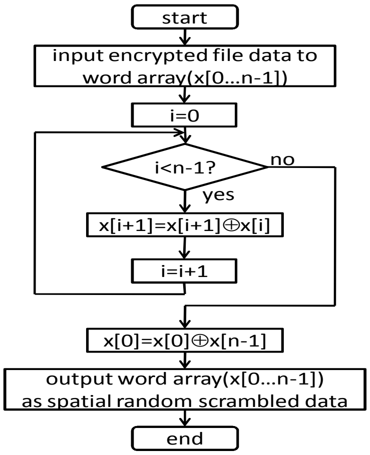
start
input encrypted file data to word array(x[0...n-1])
i=0
i<n-1?
no
yes
x[i+1]=x[i+1]⊕x[i]
i=i+1
x[0]=x[0]⊕x[n-1]
output word array(x[0...n-1]) as spatial random scrambled data
end

Figure 5. Algorithm of spatial random scrambling

In this case, a reversible operation (three kinds of operation being exclusive-or, binary addition or subtraction) for each successive word is executed for the entire file. This computation process should be repeated several times. The kind of operation can also be changed each time. The selection of operations and the number of repetitions are kept as metadata. It is strongly recommended to repeat this process several times. To de-scramble, it is only necessary to perform the same operations in the reverse order. By introducing the above mentioned spatial random scrambling technology, deciphering by a third party by comparing the encrypted fragments and the original plain text becomes almost impossible. That is, deciphering the original data become almost impossible, because of the random distribution of fragments introduced by this spatial random scrambling.

2) Fragmentation

One of the innovative ideas of this backup mechanism is combination of fragmentation and distribution in random order. Even if a cracker captured all raw packets between data center and client nodes, it would be tremendously hard to assemble all the packets in the correct order, because it would be necessary to try about (no. of fragments)! attempts.

Furthermore, the proposed backup mechanism duplicates each fragment and encrypts all fragments with different encryption keys. Therefore, no one can identify even one encrypted fragment and it is not possible to identify the other encrypted fragments. Crackers would require innumerable attempts to decipher. Since in a block cipher, the data is divided into a number of blocks, the processor and memory cost increase exponentially when the data volume increases. However, in a stream cipher, all input data are operated on bit by bit using a simple arithmetical operation, and high-speed processing can be easily achieved. Figure 6 explains the qualitative cost/performance comparison with the conventional encryption system.

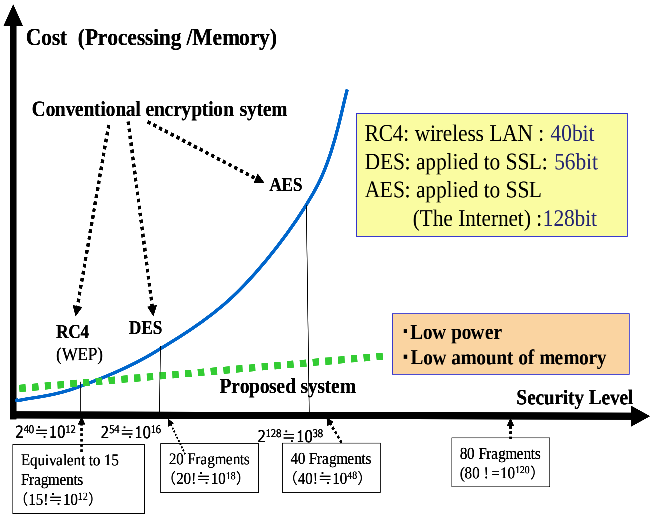

Figure 6. Cost/performance comparison with conventional encryption system

3) Duplication

The main purpose of duplication is to make possible to improve the probability of recovery. The probability of recovery rate can be estimated from the following equation. Here, P is the failure rate of each client node, n is the degree of duplication of each fragment, and m is the number of fragments.

$$\text{Probability of recovery} = (1-P^n)^m \cong 1-mP^n$$

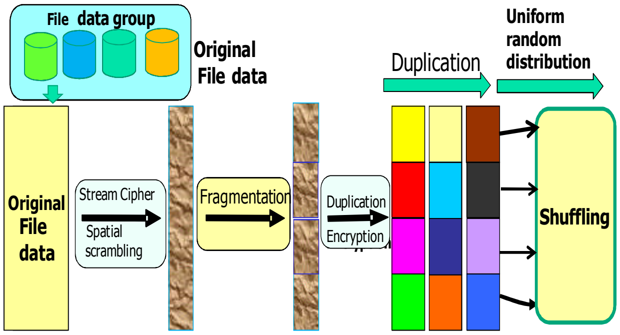

Figure 7. Sequence of a series of data processing

As mentioned above, the degree of duplication is quite important for ensuring a rigid security level, since the recovery failure rate is dependent on P^n. After sequential random scrambling, a data center divides the file data into a number of fragments. In addition, it records the data size and offset of all fragments in "encryption meta data".

Figure 7 shows the sequence of processing such as "encryption by stream cipher, fragmentation, duplication and second encryption".

As shown in Figure 8, whenever the original data is processed in the data center, then the corresponding items of metadata (Metadata#1, Metadata#2, Metadata#3) are produced sequentially. The size of this metadata is much smaller than the original data, in the order of 10^{-4} or 10^{-5} times smaller. However, it needs to be secretly and accurately transferred to the several supervisory centers in order to decipher the original data in preparation for the situation where a disaster occurs.

Figure 9 shows the calculation results of the file recovery rate, on the condition that each file fragment's failure rate P is assumed to be 0.2. For example, if the original file is divided into 30 fragments, and 30 copies are made of each fragment, then, the recovery failure rate becomes less than 10^{-19}, which is commercially available high reliability.

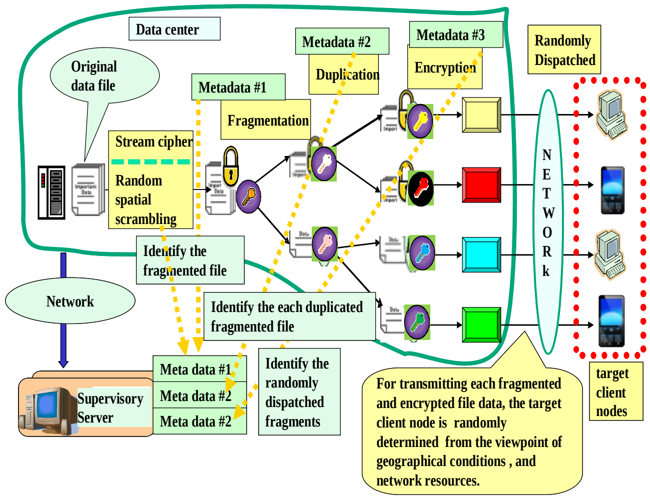
Data center
Original data file
Metadata #1
Metadata #2
Metadata #3
Fragmentation
Duplication
Encryption
Randomly Dispatched
Stream cipher
Random spatial scrambling
Identify the fragmented file
Identify the each duplicated fragmented file
Network
N E T W O R k
Supervisory Server
Meta data #1
Meta data #2
Meta data #2
Identify the randomly dispatched fragments
For transmitting each fragmented and encrypted file data, the target client node is randomly determined from the viewpoint of geographical conditions , and network resources.
target client nodes

Figure 8. Metadata transfer to the supervisory centers

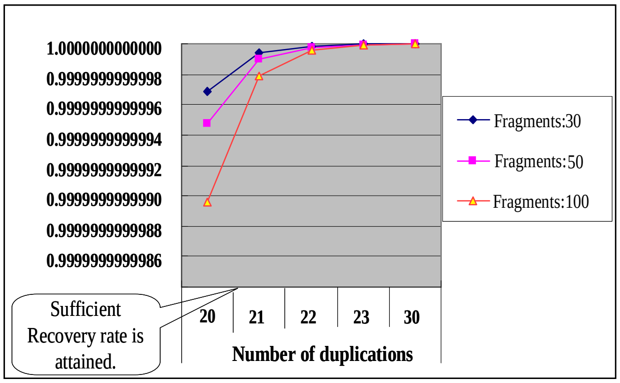

Figure 9. File recovery rate characteristics

4) Distribution

From a disaster recovery point-of-view, the duplicate copies of fragments are distributed over a wide area, in preparation for the case where a disaster is concentrated in a small area. To achieve this, the proposed file backup mechanism classifies client nodes into certain logical groups. The same logical group contains several client nodes in different geographical locations. One set of duplicated fragments which have been derived from the same fragments have to be distributed to several client nodes in some logical groups since each fragment is sent to more than one client node in each logical group because of the risk of some nodes not being alive when recovery is required. This rule ensures a secure geographical distribution, as far as possible. It is very important to ensure the uniformity of distribution, and the details of this are discussed in Section V.

D. *Characterisutics of the required funcitionss*

In terms of file data fragmentation, a rough estimate of the security level when the proposed data backup technology is used may be obtained as follows.

Case (1)

If a data file is divided into 40 fragments which are distributed to various client nodes, the number of combinations to be checked in order to reassemble the file is about $40! \fallingdotseq 2^{160}$.

This is equivalent to the AES (128 bits) cipher code.

Case (2)

If a data file is divided into 80 fragments which are distributed to various client nodes, the number of combinations to be checked in order to reassemble the file is about $80! \fallingdotseq 2^{400}$.

This is more secure than the current AES (256 bits) cipher code and such reassembly cannot be effectively realized at present.

Case (3)

When a data file is divided into 80 fragments which are duplicated to 10 copies, the number of combinations to be checked in order to recompose is about ${}_{800}P_{80} \fallingdotseq 2^{770}$. This is more secure than with the AES (512 bits) cipher code and such reassembly cannot effectively be realized in the near future. On the other hand, the number of duplications should be carefully determined from the viewpoint of the data recovery rate and the client node failure rate and the required recovery rate of each file fragment.

In terms of file data duplications, a rough estimate of the recovery rate when the proposed data backup technology is used may be obtained as follows.

Case (1)

If an acceptable file capacity for each client node is assumed to be 5 Mbytes, the original 100 Mbytes file needs to be fragmented into at least 20 files. If the duplication degree of each file fragment is 10 and each client node's failure rate is 0.2 at worst, then the probability that the original file can be recovered is assumed to be about 0.999998, which is a very high recovery probability.

Case (2)

Let us consider the case that the file data is fragmented into 80 files. When the duplication degree of each file fragment is 10 and each client node’s failure rate is 0.2 at the worst, then the probability is assumed to be about 0.999992.

V. UNIFORMITY OF DISTRIBUTION TABLE

A. *Background*

When we distribute the data of the disaster recovery system to widely dispersed sites, an appropriate distribution table must be calculated. When we calculate the great number of the table, if the entire set of the distribution tables is biased, there exist weak points in the corresponding recovery system. To avoid these problems we examined two shuffle algorithms and two random number generators.

B. *Shuffle algorithms*

At first, we use the shuffling method "Simple shuffle"; here RNG means some random number generator.

```
for j := 0 to m - 1 do d[j]:=j;
for j :=0  to m-1 do
begin
  a:=RNG mod m;
  b:=j mod m;
  x:=d[a];
  d[a]:=d[b];
  d[b]:=x;
end;
```

Next, we use the shuffling method "Fisher-Yates shuffle" as follows [23][24]. The Fisher--Yates shuffle is unbiased, so that every permutation is equally likely.

```
for j := 0 to m - 1 do d[j]:=j;
for j := m-1 downto 0 do
begin
  a:=RNG mod (j+1);
  b:=d[j];
  d[j]:=d[a];
  d[a]:=b;
end;
```

C. *Pseudo random number generators*

We selected two random number generators. One is MT (Mersenne twister), and the other is the additive random number generator as follows.

```
arrn:=100;
adata:array[0..arrn-1] of LongWord;
rct:integer;
procedure addRandomize;
var
  i:integer; x:LongWord;
begin
  Randomize;
  rct:=0;
  for i := 0 to arrn-1 do
    adata[i]:=Random(65536)+(Random(65536)shl 16);
end;

function addRandom(m:LongWord): LongWord;
var
  a:integer;

begin
  rct:=rct mod arrn;
  a:=(arrn+rct-1) mod arrn;
  adata[rct]:=adata[rct]+adata[a];
  Result:=adata[rct] mod m;
  inc(rct);
end;
```

Note that we can realize fast implementation for the additive random number generator in if $arrn = 2^k$.

D. *Goodness of fit testing*

By using MT or the additive number generator as RNG, and in addition, by using Fisher-Yates shuffle or Simple shuffle as shuffling algorithms, and setting the division number to be 100, we calculated the distribution table 10^8 times for each method. We counted the frequency of the array index which stored the sampling number 50from 100 samples, for example.

For example, under the method of MT and Fisher--Yates shuffle of 3 cycles with division number 100, we observe the sampling number 50, and count up the frequency distribution as follows.

999517,1001965,999602,999831,1000230,999741,1000156,999623,999613,999519,1001023,999375,999689,999980,997614,1002169,1000105,999576,1000608,1001074,1000407,1000744,999558,998516,1002049,1000106,999233,1000582,999225,999697,999109,999216,999520,1000540,100056,999752,999012,1002717,1000808,999358,999477,998880,1000632,999688,999596,999843,998754,998856,999658,1000298,999671,999637,1000480,1000065,1001297,1001273,1001072,999980,1000616,1000838,1000892,999637,997561,999888,1000670,1001821,1001103,1000546,999836,1000647,998212,1000111,998973,999074,1000025,998321,1000037,1000016,999397,999210,1000867,998826,1000179,1001797,1000146,1000740,1000043,1000651,1001629,999275,999771,999762,999028,1000959,999904,999932,999305,999523,1000119,999236.

The null hypothesis for goodness of fit testing is set as follows.

H_0: These frequency tables have uniform distribution.

When we divide the data to be backed up into 100 fragments, the results of the significance level are shown in TABLE I. Here, "MTFYS3r" means that RNG is Mersenne Twister and shuffle is Fisher-Yates with 3 rounds, and "addSimp3r100" denotes that RNG is additive, shuffle is Simple with 3 rounds with the condition that arrn=100.

TABLE I. SIGNIFICANCE LEVEL COMPARISON

MTFYS3r	MTSimp3r	MTFYS1r
0.8416	0.6746	0.4617
addFYS3r100	addSimp3r100	addFYS1r256
0.842	0.03372	0.2877

In calculating a distribution table for the disaster recovery system, we should use Mersenne Twister as the RNG and the Fisher-Yates shuffle as the shuffling algorithm. If we use the additive random number generator, we should use the Fisher-Yates shuffle with 3 rounds.

VI. SECRET DECENTRALIZATION OF SUPERVISORY CENTER

As only supervisory servers contain the “encryption metadata”, the supervisor servers must also take account of an unexpected natural disaster, a cyber-terrorism attack, or information leakage.

One useful solution is to set up several supervisory servers in different geographical locations, and to ensure that all of the supervisory servers have the same copy of the “encryption metadata”.

This solution is acceptable for protection from a natural disaster, but it increases the possibility of security incidents. For instance, all the backup data could be stolen if only one supervisory server was cracked.

As an alternative, the proposed solution for the distributed file data back-up mechanism has the following distinctive features as follows.

1) The separation of the “encryption metadata” database (DB) and the alive/valid information DB.

2) The introduction of a secret sharing scheme in the data center.

The supervisory server for maintaining the keep alive or valid information DB has to be Internet reachable and must always be waiting for keep alive packets from all client nodes to ascertain that some of them are valid..

On the other hand, the supervisory server for “encryption metadata” DB usually communicates only with the data center.

Therefore, the selective and effective separation of several DB servers is reasonable.

We should clearly take the different logical functions assigned to each DB into account. For instance, alive information DB servers only have lists of client nodes, and crackers are unable to decrypt the backup data in client nodes.

In addition, since “encryption metadata” DB servers are usually not Internet reachable, no cracker can intrude directly, except for an insider.

The “encryption metadata” size is about m×n×(length of encryption keys), when a data file is divided into m fragments and n copies of it are made. This means that the resource cost and maintenance cost of realizing the secret sharing scheme with “encryption metadata” will be quite low.

As a result, it is appropriate that the proposed system introduces the secret sharing scheme in the data center. Before sending “encryption metadata” to the supervisory servers, the data center processes the secret sharing scheme and creates some functional shares. Then, the data center sends each shared information file to several different supervisory servers. After the distribution of duplicated fragments and the sending of shared information files, it is quite difficult to find out a series of “encryption metadata” by itself in the proposed system.

From a disaster recovery point of view, the secret sharing scheme with some appropriate thresholds should be introduced in the proposed system. If the system uses a (3, 5) -threshold scheme, the system needs five supervisory servers, and the system can tolerate the simultaneous failure of two servers.

On the other hand, from a cyber terrorism point of view, if the system uses a (3, 5) -threshold scheme, a cracker has to intrude at least three “encryption metadata” servers and one alive/valid information server at the same time.

The configuration of the secret decentralized supervisory servers and data center is shown in Figure 10. As the proposed system uses a (2, 3)-threshold scheme, there are three “encryption metadata” share servers.

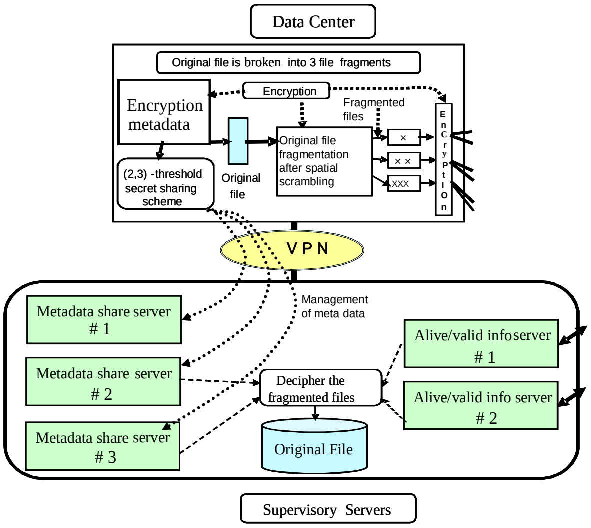
Data Center
Original file is broken into 3 file fragments
Encryption metadata
Encryption
Fragmented files
Original file fragmentation after spatial scrambling
×
× ×
xxx
EnCryPtIOn
(2,3) -threshold secret sharing scheme
Original file
VPN
Management of meta data
Metadata share server # 1
Metadata share server # 2
Metadata share server # 3
Decipher the fragmented files
Original File
Alive/valid infoserver # 1
Alive/valid info server # 2
Supervisory Servers

Figure 10. Configuration of secretly decentralized supervisory servers

As a result, there are two “keep alive” information servers. It is also possible that the Data center adopts the function of the secret sharing scheme. This unit creates some shares from the “encryption metadata”. Anyone who has to recover the backup-up file data must request the corresponding recovery to the administrators of the appropriate supervisory servers. As shown in Figure 10, at least two administrators of “encryption metadata” share servers, and one administrator of “keep alive” information servers should be required at the same time. When the administrators receive the recovery request, they gather their shared DBs and alive and valid information at the same time. If the number of shares exceeds the threshold, they can reconstruct the “encryption metadata”. Then, the normal recovery process starts.

VII. Performance Evaluation for Realizing Required Functions in the Proposed System

This section discusses the encryption performance and the spatial scrambling performance of the proposed disaster recovery system. We examined three systems for performance evaluation. TABLE II describes the test environment of each system.

TABLE II. SYSTEM ENVIRONMENTS

	Computer A	Computer B	Computer C
CPU	Core2 Quad Q6600 2.40GHz		Atom Z530 1.6GHz
memory	2GB(DDR2-800)		1GB(DDR2-533)
HDD	SATA 250GB 7200rpm	RAID 0(striping) SATA 500GB 7200rpm x4	IDE 40GB 4200rpm
OS	Fedora 10		
Kernel	2.6.27.5-117.fc10.i686.PAE		
gcc	gcc (GCC) 4.3.2 20081105 (Red Hat 4.3.2-7)		
libc	glibc-2.9-2.i686		

On these three systems, we tested the following data processing sequence.

(1) File I/O read process
Test program reads the original file from disk.

(2) Encryption process
It encrypts the whole of the read data using a stream cipher in memory.

(3) Spatial random scrambling process
It performs spatial random scrambling on the encrypted data, six times in memory.

(4) File I/O write process
This program writes the encrypted and scrambled data to disk.

We examined five file sizes, namely 64MB, 128MB, 256MB, 512MB and 1024MB.

Exceptionally, we omitted the 1024MB file from Computer C, because of main memory size restriction. We carried out the execution of the test program five times on each file, and thereafter evaluated the mean processing time to provide the results.

TABLE III shows the performance of encryption and spatial random scrambling.

On Computer A and Computer B, the software based encryption achieved better performance than Gigabit Ethernet class throughput. When the corresponding backup data is processed, it can be sent/received effectively by the Gigabit Ethernet class speed media. We confirmed that we can realize software based encryption while receiving the original data via a Gigabit level network interface. It was also ascertained that the low-end Atom system was able to achieve about half the performance of Gigabit Ethernet throughput with our software based encryption.

TABLE III. ENCRYPTION AND SPATIAL RANDOM SCRAMBLING SPEED

File size	Computer A [MB/sec]	Computer B [MB/sec]	Computer C [MB/sec]
64MB	203.7	203.0	73.20
128MB	203.2	202.9	72.84
256MB	203.4	203.1	73.00
512MB	203.2	203.0	74.63
1024MB	203.3	202.5	

TABLE IV and Figure 11 show the evaluation results for examining the above mentioned four individual processing steps for Computer A. In the TABLE IV, the different rows correspond to different original file sizes, and each column shows the time taken for each processing step.

In the Fig. 11, the X-axis of the graph shows the five file sizes, and Y axis means the total elapsed time including of four processing steps. The each elapsed time for different step can be identified by four colored areas. The blue colored area in the bottom stands for the file read, the red colored area, the 2nd from bottom, stands for the encryption, the green colored area, the 3rd from the bottom stands for the spatial random scrambling, and the purple colored area, on the above, stands for the file write.

TABLE IV. PERFORMANCE EVALUATION FOR COMPUTER A

Steps File size	file read [sec]	Encryption [sec]	Scrambling [sec]	file write [sec]
64MB	0.1102	0.04898	0.2652	0.1463
128MB	0.2130	0.09848	0.5315	1.386
256MB	0.4129	0.1961	1.063	3.418
512MB	0.8257	0.3922	2.127	8.138
1024MB	3.178	0.7846	4.252	17.66

It is ascertained that in case of Computer A, the disk I/O consumes about 80 percent of total elapsed time. In particular, the file write processing step consumes most of processing time. It is also confirmed that the file write performance of the 1024MB file is about 58 MB/sec. This result is much the same as the consumer electronics HDD average performance. Therefore, it is assumed to be better to introduce an SAS drive that has a rotational speed higher than 10k rpm, RAID-0 (striping), or Solid State Disk Drive. TABLE V and Figure 12 show the evaluation results for Computer B. Computer B in which the HDD is RAID 0 configuration, showed a performance improved over that of computer A.

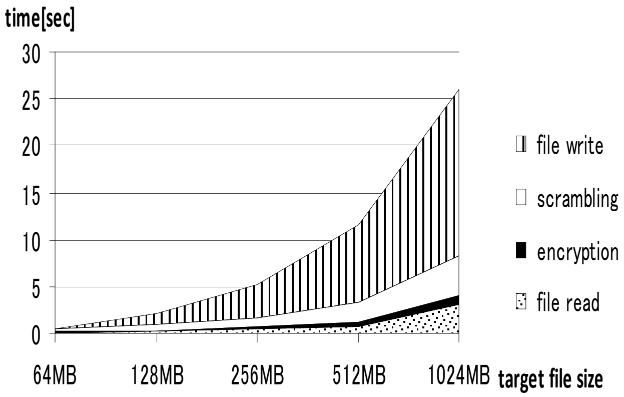

Figure 11. Processing time (v. file size) on Computer A

In TABLE V, the file write performance of the 1024MB file is about 350 MB/sec, that is about six times faster than the HDD of Computer A.

TABLE V. PERFORMANCE EVALUATION FOR COMPUTER B

Steps File size	file read [sec]	Encryption [sec]	Scrambling [sec]	file write [sec]
64MB	0.1090	0.04914	0.2661	0.1659
128MB	0.2274	0.09878	0.5321	0.3365
256MB	0.4841	0.1967	1.064	0.6694
512MB	1.311	0.3946	2.128	1.385
1024MB	4.022	0.7982	4.259	2.955

Considering these results, the Disk I/O performance is the most important factor in the software based encryption processing system. If we should encrypt a huge file such as one larger than 100GB, it would be difficult to process the encryption and spatial random scrambling in memory. In such cases, the encryption and the spatial random scrambling would have to be done after buffering and making use of multiple disk read/write operations.

TABLE VI shows the results of total encryption processes time with multiple I/O operations within a single disk and with single I/O operation in a disk for both Computer A and Computer B. In the case of the multiple I/O operations, in the encryption processing step, the whole file is read and written as is the case with a single spatial random scrambling step process. Therefore, the total elapsed time for the encryption with multiple I/O operations within a single disk includes 1 encryption time, six spatial random scrambling times, and seven disk read and write times.

In the TABLE VI, the columns of multiple I/O operations within a single disk show the total encryption

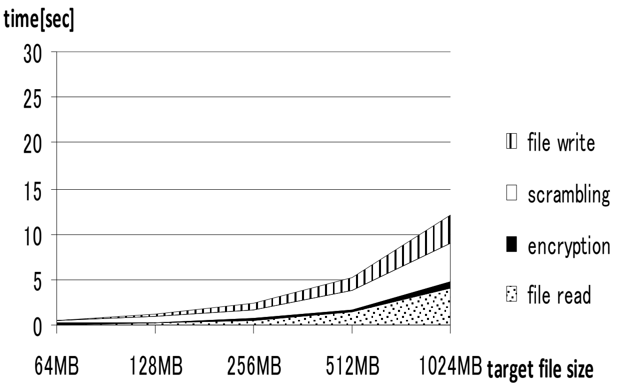

Figure 12. Processing time (v. file size) of Computer B

time using a 1MB buffer. The columns of single I/O operations in a disk show the total encryption time, read from TABLES IV and V. The total encryption time for multiple I/O operations in a single disk is five to six times greater than the total encryption time using a single Disk I/O operations on Computer A.

The total encryption time using multiple I/O operations in a single disk on Computer B is clearly improved over that for Computer A. However, the total encryption time using multiple I/O operations in a single disk for Computer B is still two to three times longer than that using a single disk I/O operations.

TABLE VI. TOTAL TIME TAKEN FOR ENCRYPTION PROCESSES

File size	Computer A		Computer B	
	Multiple I/O operations within a Single Disk	Single I/O operations	Multiple I/O operations within a Single Disk	Single I/O operations
64MB	5.375(s)	0.5707 (s)	1.367(s)	0.5901(s)
128MB	14.25 (s)	2.220 (s)	3.151(s)	1.195(s)
256MB	30.17 (s)	5.090 (s)	6.515(s)	2.414(s)
512MB	62.92 (s)	11.48 (s)	12.48(s)	5.218(s)
1024MB	124.8 (s)	25.88 (s)	33.77(s)	12.03(s)

From these results, it is apparent that we should reduce the disk I/O overhead especially for the encryption of the very large file. One of the effective solutions for the reduction of the disk I/O is pre-fragmentation.

The function of pre-fragmentation is to divide a huge file into some smaller size files and so to reduce the file size appropriately so that it can be handled in the main memory before the encryption and the spatial random scrambling stages. In this way, the encryption and the spatial random scrambling function can process each part of a large file in memory and with a single disk I/O. The following results can be ascertained.

(1) The software based encryption and spatial random scrambling process is very fast if we provide a sufficiently powerful CPU and enough memory.

(2) We need to reduce the disk I/O overhead when handling very large back-up files. Pre-fragmentation of the original file is assumed to be one of the realistic solutions for reducing the disk I/O overhead. Single disk I/O is unavoidable, since we have to save the original file or encrypted file to disk temporarily.

We need to take these technical points into account, when we commercialize the proposed disaster recovery system for practical use.

VIII. USER-FRIENDLY SERVICE LEVEL ASSURANCE

It is very important to provide a prompt response to the various service levels demands from users which will be changed often to ensure reliable and economical backup systems. It is desirable that the backup systems are able to provide the versatility which is demanded by changes in service level requirements, such as security strength and/or guaranteed recovery rate, by effectively utilizing the available network resources. In this case it is necessary to provide a user-friendly Web interface to realize the above mentioned requirements.

We propose an appropriate network architecture to offer the appropriate service levels that will incorporate both security strength level and recovery rate parameters as shown in the Figure 13.

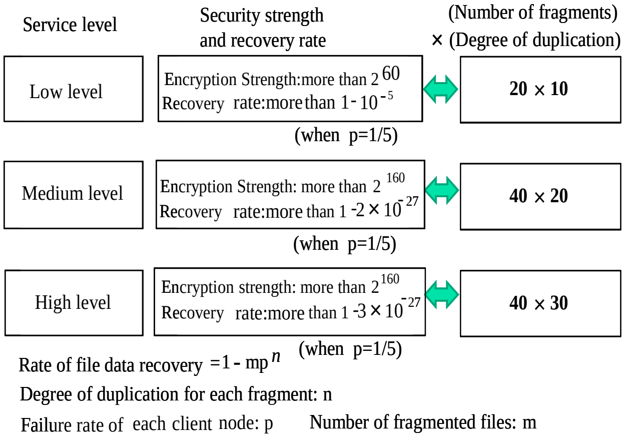
Service level
Security strength and recovery rate
(Number of fragments) × (Degree of duplication)
Low level
Encryption Strength:more than 2^{60}
Recovery rate:more than $1 - 10^{-5}$
(when p=1/5)
20 × 10
Medium level
Encryption Strength: more than 2^{160}
Recovery rate:more than $1 - 2 \times 10^{-27}$
(when p=1/5)
40 × 20
High level
Encryption strength: more than 2^{160}
Recovery rate:more than $1 - 3 \times 10^{-27}$
(when p=1/5)
40 × 30
Rate of file data recovery $= 1 - mp^{n}$
Degree of duplication for each fragment: n
Failure rate of each client node: p
Number of fragmented files: m

Figure 13. Disaster recovery service levels

The configuration of the service level control system is shown in Figure 14. The system consists of an RFID tag reader, the backup data storage PC of the user agent, a Web server which accepts user demands regarding the backup service level, and the Data Center which distributes the encrypted and fragmented backup file data to widely distributed client nodes including mobile PCs [25].

As the practical and highly secure means of managing user-friendly service levels requests, the RFID tag system including user ID, passwords and service levels which will be provided by the service providers is utilized effectively. In addition, a Web server which accepts the use requests and certifies user identification before informing the corresponding user requests to the Data center. It can specify one of several service levels.

Compared with the conventional method, whereby the user ID and password are input directly to the Web server by users, and which is also vulnerable to hacking by, for example, SQL injection, the above mentioned proposed access control method can safely realize ID certification using RFID tag information even if a lot of users access the Web server at the same time.

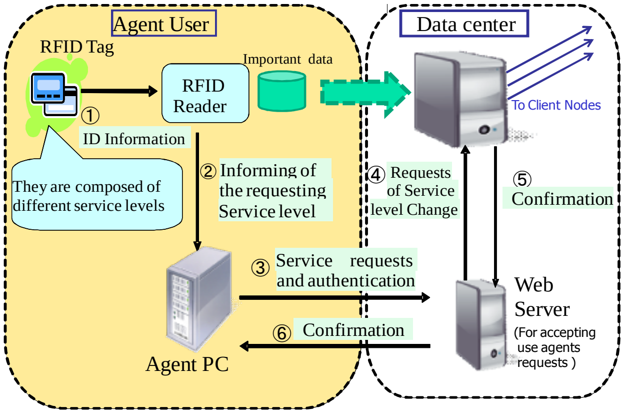
Agent User
RFID Tag
Important data
RFID Reader
①
ID Information
They are composed of different service levels
② Informing of the requesting Service level
Agent PC
③ Service requests and authentication
⑥ Confirmation
Data center
To Client Nodes
④ Requests of Service level Change
⑤ Confirmation
Web Server
(For accepting use agents requests)

Figure 14. Disaster recovery service level control architecture

In this case it is important to evaluate the average response time from the WEB server to a request from a user agent by taking both the number of simultaneous access attempts and the memory usage rate in the data center into account. The Web server frequently accesses the data center to ascertain whether the corresponding user requiring service level change is acceptable or not by considering available network resources. Generally the data center handles functions such as data encryption, fragmentation, distribution, metadata generation, etc. and so requires a much higher memory usage rate than the Web server, so we first evaluated the memory load conditions of the data center as the parameter to be studied, as shown in Figure 15.

In this case, we used an Athlon™ 643200 as the processor in the data center with 1GBytes of RAM. For the Web server, we used Celeron(R)(1.7GHz) with 512MBytes of RAM.

We evaluated the average response time per message of the Web server to ensure the user's convenience when the memory of the Data center is highly loaded (from about 94% - 95%). Consider the case where the number of user agents using the file data back up service is 1000, and 10% of users request a change in service level at the same time, as the worst cases. Taking these conditions into account, we assumed that the required request message handling capacity will be sufficient to provide 100 messages/s in both the Web server and the data center server. The experimental user request message generation rate (50~200 messages/s) was based on these conditions.

Figure 15 shows the evaluation result of the average response time of the Web server when 50~200 messages per second need to be handled at both the Web Server and the data center server. It was confirmed that the average response time of the Web server increases rapidly when the memory usage rate of the data center is 94.4% under all request message generation conditions. According to the experimental data, we found that the memory processing load in the data center should be less than 94% to ensure a prompt response time and corresponding user convenience.

Both servers can be logically separated, but can also be physically integrated into an unified server. In this case the above mentioned experimental results will still apply. From the viewpoint of security, the servers should be deployed in geographically different sites.

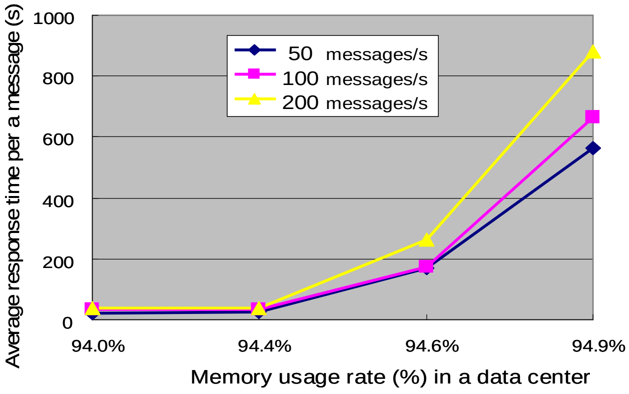

Figure 15. Average response time per message

IX. EXAMPLE OF A SUBSTANTIAL SYTEM

The proposed system architecture can be used in a real-time secure video data monitoring service such as that of a security service company. The main features of the prototype system which we produced are as follows.

The prototype system is mainly composed of two parts as shown in Figure 16. One includes the monitoring camera (SANYO VCC-P450, H. 264), the DRT Processor (Atom N270, 1.6GHz, 1Gbyte RAM), which includes the functions of data division, spatial scrambling, encryption, and distribution and the gateway to the Internet, all deployed in the observed site. Here, DRT stands for the proposed "Disaster Recovery Technology". The interface speed between the camera and the DRT processor is 500kb/s at maximum and the gateway has a 100 Mb/s optical fiber interface. The other part has the functions of data collection, decryption and storing which are implemented in several storage servers deployed in the several surveillance centers. In the minimum configuration at the surveillance center, the DRT processor can also include the function of storage server.

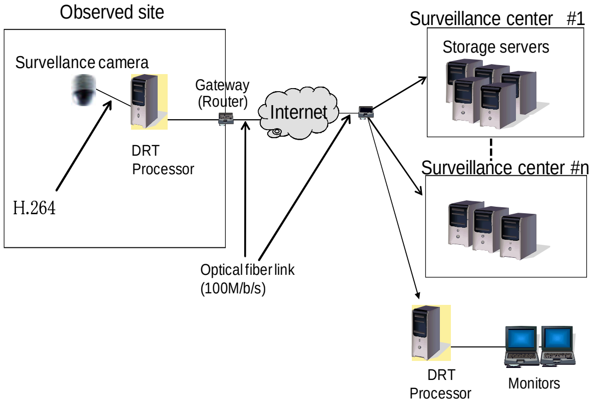

Figure 16. Example of a introduction in a practical system

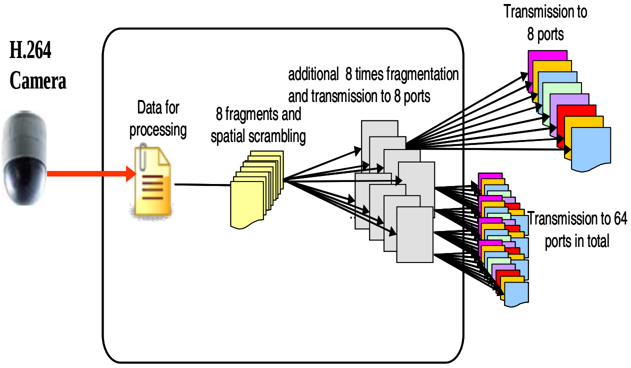
H.264 Camera
Data for processing
8 fragments and spatial scrambling
additional 8 times fragmentation and transmission to 8 ports
Transmission to 8 ports
Transmission to 64 ports in total

Figure 17. Detailed configuration of the proposed system

The DRT processor at the observed site takes five pictures per second from the camera and transmits them continuously to the surveillance center via the Internet in real time. The size of compressed pictures in the surveillance center is about 40 K Bytes in VGA size. For efficient processing and transmission in the Internet, we fragmented the 40~50Kbytes data into 64 fragments, each of about 1 Kbyte. These fragmented data can be distributed to several storage servers in the different surveillance centers.

In Figure 17, the original data is fragmented into 64(=8 x 8) fragments by adopting 2 stages of fragmentation. Actually, in this case of the prototype DRT processor, it fragments the corresponding pictures into eight fragments and executes spatial scrambling and thereafter, sends them to 64 logically separated ports (storage servers) after an additional eight times fragmentation of each fragment.

After the spatial scrambling of the set of eight data file fragments, each file can be uploaded to eight different servers after the additional further fragmentation. In the prototype system, the corresponding fragments can be uploaded to at most 64 different servers. In the prototype described here, we did not produce duplicated fragments, for reasons of network control simplicity and to increase real time performance.

In addition, each fragment has a delivery ID which corresponds to the secret key and is created by the combination of the specific DRT processor number and IP address. The delivery ID is commonly used between the surveillance center and the observed site. In the supervisory center, the original data can be recovered by making use of the delivery ID and the corresponding password. In order to determine the destination server number to which each fragment should be uploaded, we temporarily used the random number generation “rand( )” function. The function rand( ) is called with initial value for the pointed object. Ideally, we should use the method explained in the previous Section IV. This method will be introduced in the commercial products.

When the number of users and corresponding cameras and video information increase, the number of servers should be increased and the number of fragments should be increased to ensure the firm and rigid security. This will be a topic for future study.

X. Conclusion

In this paper, we have proposed an ultra-widely distributed file backup mechanism implemented by effectively combining a series of technologies [KF It seems unnecessary to call all of these ‘technologies’ every time, so I have simplified this here.]including an effective stream cipher code in each encryption and distribution stage, spatial file scrambling, the random distribution of file fragments and the deployment of several secure supervisory servers . By making use of widely spread PCs, PDAs and cellular phones, a system for prompt and secure file backup can be realized economically. By making use of the proposed mechanism, illegal data recovery by tapping by a third party becomes almost impossible and an extremely safe and economically viable data back up system can be realized.

We clarified that we should use Mersenne twister as the RNG and the Fisher-Yates shuffle as the shuffling algorithm. If we use the additive random number generator, we should use the Fisher-Yates shuffle with 3 cycles.

We clarified that the proposed software based encryption achieves better performance than Giga-bit Ethernet class speed, and that a low-end Atom system is able to achieve about half the performance of Giga-bit Ethernet throughput. It has been found that reducing the disk I/O overhead for the encryption of very large files is very important and an effective solution for the reduction of disk I/O is pre-fragmentation.

We have also proposed a network architecture which can realize a user-friendly service level control mechanism using an RFID tag-reader and a corresponding Web server interface. We have demonstrated secure protocols to change a user’s required service levels. In addition, we have evaluated the average response time from the Web server by taking the memory usage rate of the data center and an appropriate number of simultaneous request messages per second into account.

We have implemented a practical disaster recovery system by using the proposed technology and clarified the system specification.

As for future research we should take into account narrow bandwidth and unreliable connections with the huge amount of communication devices in order to avoid possible inconsistency of gathered fragmented data, considering that the backup data can be dynamically updated. And in addition, an optimum network utilization technology should be introduced.

We are planning to verify the key features to fully utilize the network resources for ideal commercialized disaster recovery systems.

ACKNOWLEDGMENT

This work has been partially supported by the JST Innovation Bridge (Issue number: 08030256), Grants-in-Aid for Scientific Research (Issue number: 21560414), and Tokyo Denki University Research Center for Advanced Technologies (Q07J-13). This work was also supported in part by a grant of Strategic Research Foundation Grant-aided Project for Private Universities from Ministry of Education, Culture, Sport, Science, and Technology, Japan (MEXT), 2007-2011(07H012).

We also greatly appreciate the continuous support of all the staffs of Tokyo Denki University Center for Research Collaboration (TLO accredited).

REFERENCES

[1] N. Miyaho, Y. Ueno, S. Suzuki, K. Mori and K. Ichihara, “Study on a Disaster Recovery Network Mechanism Using Widely Distributed PC Client Nodes,” ICSNC2009, pp.217-223, Sep., 2009.

[2] S. Suzuki, “Additive cryptosystem and world wide master key,” IEICE Technical Report, 101(403): pp.39‐46, ISEC2001-84.Nov., 2001.

[3] N. Miyaho, S. Suzuki, Y. Ueno, A. Takubo, Y. Wada, and R. Shibata, “Disaster recovery equipments, programs, and system,” Patent publication 2007/3/6 (Japan), PCT Patent :No.4296304, Apr., 2009.

[4] N. Miyaho, M.Onuki and S.Kurokawa, ”Network data distributed system” JP2007/054234, Japan Patent 2007-326288 (pending), Dec., 2007.

[5] Y.Ueno, N.Miyaho, and S.Suzuki, ”Disaster Recovery Mechanism using Widely Distributed Networking and Secure Metadata Handling Technology,” Proceedings of the 4th edition of the UPGRADE-CN workshop, Session II: Networking, 45-48, June, 2009.

[6] K. Kokubun, N. Miyaho, S. Suzuki and Y. Ueno, “Disaster recovery system performance evaluation by making use of Grid computing technology”, IEICE Network system research paper, Dec., 2007.

[7] K. Kokubun, N. Miyaho, S. Suzuki and Y. Ueno, “The evaluation of the disaster recovery system using Grid computing technology”, IEICE Technical Report NS2007-106 Dec., 2007.

[8] K. Kokubun, N. Miyaho, S. Suzuki, Y. Ueno and Y. Ohno, “Study on a disaster recovery system using Grid computing technology,” IEICE Communication Society Symposium, BS-5-10, 2007.

[9] Y. Ohno, T. Iwamoto, N. Miyaho, K. Kokubun, S. Suzuki and R. Shibata, “Study on a Disaster recovery system using Grid Computing Technology,” IEICE National Convention, BS-6, 2007.

[10] K. Kokubun, T. Iwamoto, N. Miyaho, S. Suzuki, Y. Ueno and R. Shibata, “Study on a disaster recovery system using Grid computing technology,” IEICE National Convention, BS-7-193, 2007.

[11] Y. Someya, Y. Ueno, S. Suzuki, K. Mori and N. Miyaho, ” Study on a disaster recovery system by making use of GRID computing technology,” 2008 IEICE National convention, Communication Society Symposium, BS-7-1, 2008.

[12] NTT-East “Wide area disaster recovery services”, <http://www.ntt-east.co.jp/business/solution/security/dr/>06.01.2010.

[13] Y. Kitamura, Y. Lee, R. Sakiyama and K. Okamura, ”Experience with Restoration of Asia Pacific Network Failures from Taiwan Earthquake”, IEICE TRANS.COMMUN.,VOL .E90-B, NO.11, pp.3095-3103, Nov., 2007.

[14] S. Kurokawa and N. Miyaho, “Study on the Distributed Data Sharing Mechanism with a Mutual Authentication and Meta Database Technology”, APCC2007, Oct., 2007.

[15] J. Yamamoto, M. Kan and Y. Kikuchi, “Storage based data protection for disaster recovery”, *J. IEICE*, 89(9):801‐805, Sep., 2006.

[16] A. Farley, S. Zhao, and V. Lo, “Result verification and trust based scheduling in peer to peer grids”, IEEE International Conference on peer -to-peer Computing, pp.31-38, Aug., 2005.

[17] K. Sagara, K. Nishiki and M. Koizumi, “A Distributed Authentication Platform Architecture for Peer-to-Peer Applications”, IEICE Trans.Commun.,Vol..E88-B, No.3, pp. 865-872, Mar., 2005.

[18] Y. Deswarte, L. Blain, and J. Fabre, “ Intrusion Tolerance in Distributed Computing Systems”, “Intrusion Tolerance in Distributed Computing Systems”, â IEEE Symposium on Security and Privacy, pp. 110-121, 1991.

[19] Jay J. Wylie, M. Bakkaloglu, V.Panurangan, M. W.Bigrigg, S. Oguz, K. Tew, C. Williams, G. R.Ganger, P. K. Khosla, “ Selecting the Right Data Distributed Scheme for a Survivable Storage System”, CMU-CS-01-120, School of Computer Science, Carnegie Mellon University, Pittsbugh, PA 15213, May, 2001.

[20] J.Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W.Weimer, C. Weels, and B. Zhao, “OceanStore: An Architecture for Global-Scale Persistent Storage”, <http://oceanstore.cs.berkeley.edu/(2010.6.26), and http://oceanstore.cs.berkeley.edu/publications/papers/pdf , University of California, Berkeley, ASPLOS 2000, Nov., 2000.

[21] S. Tezuka, R. Uda, A. Inoue and Y. Matsushita, “A Secure Virtual File Server with P2P Connection to a Large-Scale Network”, IASTED International Conference NCS2006, pp.310-315, 2006.

[22] R. Uda, A. Inoue, M. Ito, S. Ichimura, K. Tago, T. Hoshi,, “ Development of file distributed back up system,” Tokyo University of Technology, Technical report, No.3, pp. 31-38, Mar., 2008.

[23] Fisher, R.A. and Yates, F. [1938]. Statistical tables for biological, agricultural and medical research. London, 1948.

[24] Knuth, Donald E. The Art of Computer Programming, Volume 2: Seminumerical algorithms., 3rd edition, Addison Wesley. pp.142-146, 1998.

[25] N. Miyaho, Y. Ueno, S. Suzuki, K. Mori and A. Takubo, “Security level network control system” Japan Patent 2008-262704 (pending), 2008.