An Automated System Recovery Using BlockChain

Suhwan Bae
Department of Computing
Soongsil University
Seoul, Korea
Shbae0213@soongsil.ac.kr

Yongtae Shin
Department of computing
Soongsil University
Seoul, Korea
shin@ssu.ac.kr

***Abstract*—** The existing Disaster Recovery(DR) system has a technique for integrity of the duplicated file to be used for recovery, but it could not be used if the file was changed. In this study, a duplicate file is generated as a block and managed as a block-chain. If the duplicate file is corrupted, the DR system will check the integrity of the duplicated file by referring to the block-chain and proceed with the recovery. The proposed technology is verified through recovery performance evaluation and scenarios.

Keywords— blockchain, disaster recoevry, system recovery

I. INTRODUCTION

In order to recover the system from the existing DR system, recovery was performed using the file that replicated the data of the fixed time[1]. In addition, data integrity was ensured by encrypting the data, but if the file itself had errors, it could not be used for recovery.

Therefore, in this paper, we propose to use block chain to ensure security and availability. In order to guarantee the integrity of the duplicate file, the proposed scheme is created as a block and registered in the block-chain. Before proceeding with the recovery, compare the information in the block-chain to determine whether the replication file is in error. If the comparison result shows that the file is defective, delete the file and import the normal duplicate file for recovery.

II. AN AUTOMATED SYSTEM RECOEVERY USING BLCOKCHAIN

This chapter proposes an automated system recovery using block-chain. In the proposed technique, recovery data generation operates asynchronously. The block contains the contents of the replicated file and is registered in the block-chain. When recovery is required, the D center verifies whether the duplicated file is abnormal by referring to the block-chain, and proceeds with recovery.

A. Physical structure

Figure 1 shows the physical structure by the proposed method.

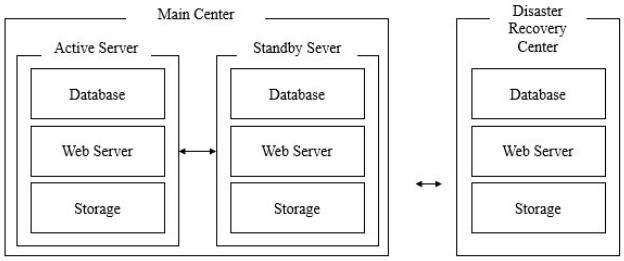

Figure 1. Physical structure in paper

The main center consists of active-standby, mirrored in real-time, and asynchronous and hot-site to the DR center. The DR center performs the functions of duplicating, storing, and restoring data in conjunction with the main center. Active servers in the main center deliver recovery data to other standby and DR center. When the recovery data is received, a block is generated. The generated block is registered in a block chain that exists in all servers and the DR center.

B. Block format[3][4]

Figure 2 shows the block format by the proposed method.

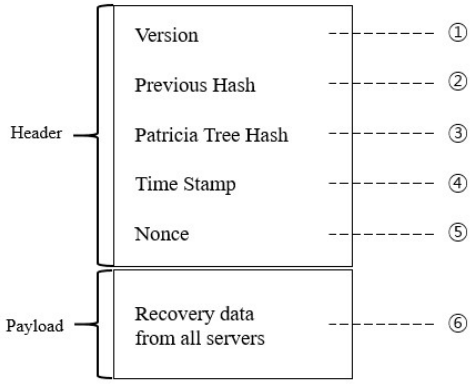

Figure 2. Block format in paper

① Specify the block-chain version currently in use

② Use the hash of the precious block to create the next block. If no previous block hash exists, use the initial vector hash.

③ Hash the contents of the payload to construct a Patricia tree, and include the root hash in the header of the block.

④ Write the timestamp when the block is created. A time earlier than the timestamp of the previous block can not be used.

⑤ Nonce is 32 bits long. When creating a block, change the Nonce to find a hash that meets the block creation criteria.

⑥ Recovery data that active servers encrypted with private keys.

C. Block creation and Registration process

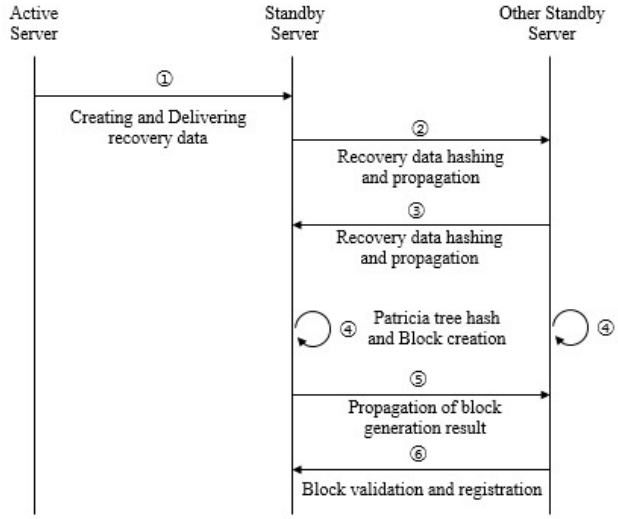

Figure 3. Block creation and registration process

① The active server generates recovery data, encrypts it with the private key, and sends it to the standby server.

② The standby server uses the SHA-256 hash to hash recovery data and passes it to the other standby server.

③ It retrieves the recovery data hash value of each server from another standby server.

④ Each Standby server creates a Patricia tree and hash root, previous block hash, timestamp. Use Nonce to generate a block by performing proofs of work

⑤ The first server to create the block informs the block creation and forwards the nonce and time stamp used to create the block to other servers.

⑥ Each standby server validates the block with the received value. Register or regenerate the block according to the verification result

D. Recovery procedures in the event of failure

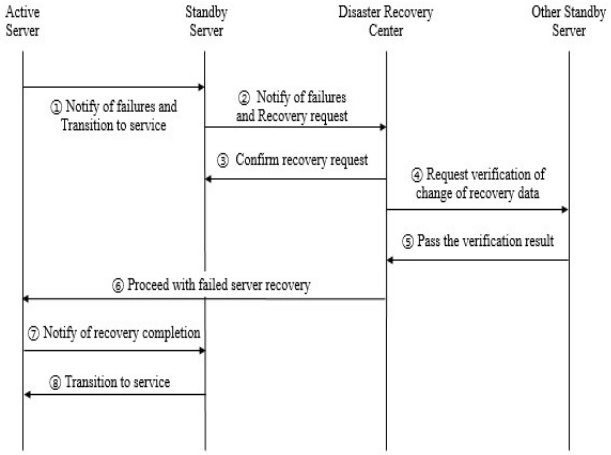

Figure 4. Failure recovery procedure

① If Active server fails, it notifies Standby server of the failure and switches service to Standby server.

② The standby server notifies the DR center about the failure and requests the recovery of the active server.

③ The DR center confirms the recovery request and sends a response to the recovery request to the standby server.

④ The DR Center sends the hash of the recovery data to other standby servers to check whether the replication file has been changed

⑤ Standby servers compare the received hash with the contents of their own block chain. If the hash matches, send a message saying everything is fine. If the hashes differ, notify that the recovery data is changed and transfer the normal recovery data.

⑥ The DR center will proceed with recovery using normal recovery data according to the verification result.

⑦ When the recovery is completed, the active server notifies the standby server of the completion of the recovery and requests the service conversion.

III. Performance Analysis

In this chapter, performance analysis of the proposed method is conducted through security evaluation.

A. Creating clone file[3]

Security assessment of recovery data is performed through two scenarios under the following conditions.

① The malicious user participates in the block generation and modifies the recovery data.

② The malicious user can no occupy 51% of the servers that generate blocks.

③ The duplicated file is registered in the block-chain by performing a Proof-of-Work[5].

When creating a duplicate file, each active server's information is sent to all the standby servers and the disaster recovery center to create the block. The malicious user changes the existing information and participates in the block creation.

The first scenario is when the malicious user first attempts to create and register a block. If the server that the malicious user is attempting to modify is 'H', the tree composition of the block is configured as shown in Figure 5.

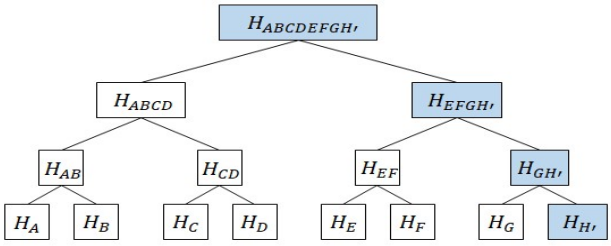

Figure 5. Tree in block created by malicious user

Since the recovery data of the 'H' server has been changed, it creates block with contents different from the configuration of the tree that other standby servers have. As a result, blocks with different contents from other blocks are created because the root hash is different. To register in a chain, the block must not be registered in the chain, since more than half of the participants in the block creation must receive the agreement.

The second scenario is that the malicious user generates blocks more quickly than normal blocks. Because the first block created is registered in the chain, it is possible to create the block before it is recognized as the normal block. The parameters used in the scenario are shown in Table 1 below.

Table 1. Security strength evaluation parameter

Parameter	Description
p	Probability that normal standby will find the next block
q	Probability that the attacker will find the next block
q^z	Probability that the attacker will quickly find the next block of z's.

The equation for the probability of an attacker catching along a normal chain is as follows.

$$q^z = \begin{cases} 1, & if\ q \le p \\ \left(\frac{q}{p}\right)^z, & if\ q > p \end{cases} \quad (1)$$

Under condition 2, an attacker can not take control of 51 % of the system and therefore becomes a p>q. As a result, the probability that an attacker catches up with the increase of the normal block decreases exponentially as the number of blocks increases. The security strength of recovery data increases gradually as an attacker can not register.

For example, if a malicious user occupies 5% of a server and all servers are of the same specification, the probability of a server forgery is shown in Table 2.

Table 2 Probability of an attacker succeeding in tamper

Number of blocks	Probability of succeeding in tamper
z = 1	0.05263
z = 2	0.00277
z = 3	1.457e-4
z = n	$(q/p)^z$

As can be seen from the above results, it can be confirmed that the probability that the attacker succeeds in the forgery is exponentially decreased.

Through the two scenarios, it is confirmed that the forged blocks are not registered in the chain and that it is difficult to continue to register the blocks even if the blocks are registered.

B. System recovery progress

The performance evaluation of the system recovery procedure was performed in two scenarios under the following conditions:

① The DR center will proceed with restoration using the duplicate files it owns.

② All the servers participating in the block creation receive the hash of the duplicate file from the disaster recovery center and proceed with the verification.

③ Restore only if the verification result is normal.

Through the two scenarios, it is confirmed that the forged blocks are not registered in the chain and that is difficult to continue.

The first scenario is to restore the system to a file that is forged by a malicious user. The DR center sends a hash of the

duplicate file to all the Standby according to condition 1 and 2 before starting recovery. Each standby server compares the received hash with the hash registered in the block-chain. As the hash of the forged file was delivered, it informs the forgery of the file and sends the normal file. Therefore, even if one duplicated file is altered up, a normal restoration is carried out.

The second scenario is to recover using a random clone file that has not undergone a normal process. Perform the procedure to verify the hash of the corresponding clone file before proceeding with the repair procedure. At this point, since no random clone file has a hash registered in the block chain, the restore request is rejected and can not be recovered.

The above two scenarios confirmed that the restore was rejected when the clone file was being manipulated or moved to a random file.

C. System recovery rate

The system recovery rate is evaluated by comparing the amount of data that can be restored during the recovery process, compared to the amount of data generated when the normal service is provided. The parameter for measuring recovery rate are as shown in Table 3.

Table 3. Parameters used in formula

Parameter	Description
D_{Total}	Amount of total data
D_{rev}	Amount of data at the time the recovery data was replicated
L_{Tottal}	Amount of data generated from the point of failure until the recovery is complete
L_{before_rev}	Amount of data that was created after the recovery point before the failure initiate recovery
$L_{rev\ k}$	Amount of data generated during k-th recovery
R_{rev}	Data recovery rate

When serviced normally, the system data are as follows:

$$D_{Total} = D_{rev} + L_{Tottal} \tag{2}$$

D_{Total} is the sum of the amount of data at the point of cloning and the amount of data created after cloning. The parameters for evaluating the recovery rate are shown in Table 3.

$$L_{Total} = L_{before\_rev} + L_{rev\_k} \tag{3}$$

$$L_{rev\_k} = \begin{cases} 0, & \text{recovery success} \\ L_{rev\ k}, & \text{recovery fail} \end{cases} \tag{4}$$

$L_{rev\ k}$ has two cases when recovery is successful or fail. However, the amount of data generated during the recovery process varies and can not be expressed in the correct formula. Total amount of data can be expressed in (5) expression until recovery is retried and successful in the event of a failed recovery.

$$L_{rev\_k} = \lim_{k\to\infty} \sum_{1}^{\infty} L_{rev\_k} \tag{5}$$

If data recovery is performed based on the above, the rate of data recovery will be shown in (6). And as the number of failed restores increases, as shown in (4), the rate of recovery decreases proportionally.

$$R_{rev} = \frac{D_{rev}}{D_{Total}} = \frac{D_{rev}}{D_{rev} + L_{Total}} \tag{6}$$

Since accurate loss data can not be calculated for performance evaluation, the following conditions are assumed and evaluated.

① D_{rev} is 100GBytes.

② L_{before_rev} is equivalent to one percent of D_{rev}.

③ $L_{rev\ k}$ is created by half as much L_{before_rev}.

Table 4 and Figure 6 are data that show recovery rates for each recovery attempt.

Table 4. Recovery rate by attempt

Attempt	Total Amount	Loss Amount	Recovery rate
1	101.5GB	1.5GB	0.9852
10	106GB	6GB	0.9434
20	111GB	11GB	0.9009
30	116GB	16GB	0.8621
40	121GB	21GB	0.8264
50	126GB	26GB	0.7937
60	131GB	31GB	0.7634
70	136GB	36GB	0.7353
80	141GB	41GB	0.7092
90	146GB	46GB	0.6849
100	151GB	51GB	0.6623

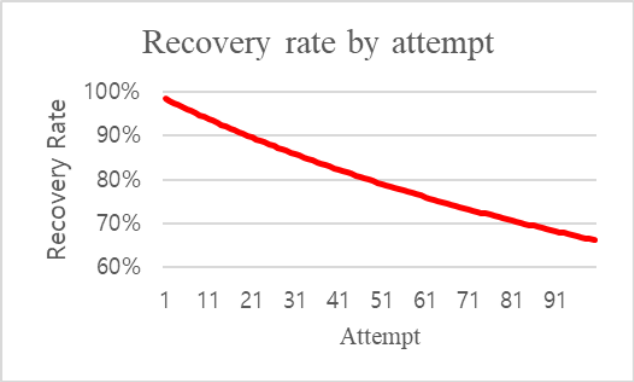

Figure 6. Recovery rate by attempt

As the data recovery attempt progresses, the amount of data loss increases. The data recovery rate also decreases. Generally, the recovery procedure is completed without multiple attempts, as in the scenario, and the amount of lost data is smaller than the amount of data at the point of recovery, which is higher than the recovery rate analyzed in the scenario

IV. Conclusion and Future work

In this paper, server recovery data is created as a block and registered in the chain. The recovery is performed by verifying whether or not the duplicate file generated for recovery is normal or not.

The performance evaluation in this study was evaluated through scenarios as a real implementation problem. As a result, we confirmed that if a duplicate file has a problem through the process of creating a block or system recovery, it can be restored normally. In addition, the recovery performance of the proposed technique was measured through the data recovery rate evaluation.

Future research will focus on researching IoT device identification, privacy protection, V2X communication, etc. by improving the content and operation procedure of block payload.

Acknowledgment

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No. 2017-0-00724, Development of Beyond 5G Mobile Communication Technologies (Ultra-Reliable, Low-Latency, and Massive Connectivity) and Combined Access Technologies for Cellular-based Industrial Automation Systems).

References

[1] IBM,“Disaster Recovery and Business Continuity version 2011”, Aug 1, 2011.

[2] KS X ISO/IEC 27001:2014 Information technology –Security techniques-Information secyruty management systems-Requirements

[3] Satoshi Nakamoto, “Bitcoin:A Peer-to-Peer Electronic Cash System”,2009

[4] Vitalik Buterin. “Ehtereum White Paper A Nect Gerneration Smart Contract & Decentralized Application Platform”. 2014.

[5] Markus Jackobosson, Ari Juels. "Proofs of Work And Bread Pudding Protocols". Kluwer Academic Publishers. pp258-272. 1999.