Article

Reinforcement Learning Based Data Self-Destruction Scheme for Secured Data Management

Young Ki Kim ¹, Saeed Ullah ¹, Kiuk Kwon ², Yunchul Jang ², Jongsoo Lee ² and Choong Seon Hong ^{1,*}

¹ Department of Computer Science and Engineering, Kyung Hee University, Yongin-si 17104, Korea; qoo0144@khu.ac.kr (Y.K.K.); saeed@khu.ac.kr (S.U.)

² Databank Systems, Daegu 705-701, Korea; kiuk.kwon@idatabank.com (K.K.); yunchul.jang@idatabank.com (Y.J.); jongsoo.lee@idatabank.com (J.L.)

* Correspondence: cshong@khu.ac.kr; Tel.: +82-31-201-2532

Received: 2 March 2018; Accepted: 23 April 2018; Published: 27 April 2018

Abstract: As technologies and services that leverage cloud computing have evolved, the number of businesses and individuals who use them are increasing rapidly. In the course of using cloud services, as users store and use data that include personal information, research on privacy protection models to protect sensitive information in the cloud environment is becoming more important. As a solution to this problem, a self-destructing scheme has been proposed that prevents the decryption of encrypted user data after a certain period of time using a Distributed Hash Table (DHT) network. However, the existing self-destructing scheme does not mention how to set the number of key shares and the threshold value considering the environment of the dynamic DHT network. This paper proposes a method to set the parameters to generate the key shares needed for the self-destructing scheme considering the availability and security of data. The proposed method defines state, action, and reward of the reinforcement learning model based on the similarity of the graph, and applies the self-destructing scheme process by updating the parameter based on the reinforcement learning model. Through the proposed technique, key sharing parameters can be set in consideration of data availability and security in dynamic DHT network environments.

Keywords: self-destructing scheme; DHT network; privacy protection; reinforcement learning

1. Introduction

According to an IDC (International Data Corporation) survey in August 2008 [1], security is one of the most challenging tasks in the cloud service. Figure 1 illustrates the results of a multi-aggregation approach to some of the factors that the cloud service should resolve. Among them, security is the most important factor, which means that it is a problem that needs to be addressed. As the number of cloud service users grows, efforts are being made to protect personal information. In addition, in order to provide amicable service, the user's data stored in the cloud is inevitably processed and stored by the system. However, general users lack the knowledge of this process, so they have difficulty managing data and there is a risk that their personal information will be leaked. In addition, personal information of the user may be leaked from the service provider's fault or attack by the hacker.

In 2009, Geambasu proposed the Vanish system as a model to protect the privacy of users of cloud services [2]. Figure 2 shows the overall structure of the Vanish system. The Vanish system utilizes existing technology and infrastructure to automatically destroy data containing personal information without additional hardware or user intervention. The main technique of the Vanish system is to use large scale distributed hash tables. The Distributed Hash Table (DHT) network is a peer-to-peer (P2P) type network composed of several distributed nodes, and VuzeDHT is used in the Vanish system.

In VuzeDHT, data has the form of index-value pair, and lookup, get and store operations are performed in the process of requesting and storing data between nodes. To retrieve a value with a specific index, the node performs a lookup operation on the node with a get operation.

Figure 1. Rate the challenges/issues ascribed to the cloud on-demand model.

Figure 2. Architecture of the vanish system.

In addition, the nodes constituting the DHT transfer data to neighboring nodes in order to increase the availability of the data.

In the Vanish system, which proposes a self-destructing scheme in which data is automatically destroyed after a certain time without any additional hardware or user intervention, the reason for using DHT as a main technique is as follows:

- **Availability.** In order to increase the availability of data, the nodes that constitute the DHT network copy the data that it has before the specified time and deliver it to the adjacent nodes. These characteristics ensure good availability of data.
- **Scale (Decentralization).** DHTs such as Vuze, Mainline, and Kademlia (KAD), which exist at present, can participate in the nodes constituting the network, that is, clients without any specific constraints. Studies on VuzeDHT and Mainline DHT assume that more than one million nodes are active at the same time [3].
- **Churn.** DHT is constantly changing as nodes constituting the network are added or deleted. This characteristic allows the data stored in the DHT to be automatically destroyed over time.

The Vanish system is designed to utilize one or more DHT networks and receives data and dead time from users and structures them into new data objects. Random symmetric key(K) is used to encrypt the data(D) received from the user and obtain ciphertext(C). Then, use Shamir's Secret Sharing [4] technique to divide key K into N pieces. The threshold(T) applied here is the minimum number of key shares needed to obtain the encryption key(K) and is determined by the user or system.

Once the key shares and thresholds are computed, an index is assigned to each key share using the random access key(L) generated by the pseudorandom number generator. The Vanish system stores each key share in the DHT network based on the generated index. At this time, depending on the type of the DHT, if the data destruction time can be specified, the information is also included. If the deployed key shares are less than the set threshold over time, the data object becomes permanently inaccessible. The newly created data object by the Vanish system consists of L, C, N, and T stored in the cloud storage and used in various forms.

In the self-destructing environment, it is necessary to set the total number of key shares and the threshold value in consideration of the environment of the DHT network in order to allow the user's data to be accessed only for a specific time and then to be automatically destroyed. However, existing studies are designed to determine this by the user or system. Typical users of cloud services will have difficulty in understanding both the structure and the principles of self-destructing systems, which is hardly a user-friendly system. In addition, due to the nature of DHT, which is changing consistently, it is somewhat difficult to satisfy the user's requirements with the number of fixed key shares and the threshold value. Therefore, in this paper, we propose a method to find the optimal threshold value to satisfy user's requirements considering the environment of dynamic DHT network in the structure of a self-destructing system.

As a method of determining the threshold value, based on the reinforcement learning model, the degree of similarity of the data availability graph over time is measured, and an optimal threshold value is found in consideration of security and availability of the data. In the proposed system, the threshold value selected as the default value in the existing system is selected as the initial value. If the DHT network does not meet user requirements as the network changes, a better threshold is chosen based on the similarity of the data availability graph. In this case, the state is defined according to the data availability graph in the environment of the DHT network, and actions for selecting the number of key shares and threshold value having different values from the existing ones. In addition, the key shares are distributed to the DHT network by finding the optimal key sharing parameters based on the similarity of the graphs as a reward.

The remainder of this paper is organized as follows. Section 2 of this paper deals with some basic background and related research, while Section 3 describes the overall system structure, graph similarity measurement method, and threshold determination process. In Section 4, we compare the proposed system with the existing system and analyze the results. Finally, in Section 5, we conclude this paper.

2. Background and Related Work

2.1. DHT Based P2P Network

A DHT based peer-to-peer (P2P) network consists of a large number of nodes, and each node can join or leave the network at any time without having a hierarchical structure. These characteristics enable nodes to share data by communicating directly within widely dispersed networks.

Figure 3 shows the structure and data processing of a distributed hash table. In a DHT based P2P system, data is associated with key value (result of hashing the data), and each node in the system handles a portion of the hash space and is responsible for storing a certain range of key values [5]. When a specific key is searched for data, the system returns the identifier (IP address) of the node storing the data with that key.

Figure 3. DHT based Peer-to-Peer (P2P) network.

The biggest advantage of such a system is that the nodes are distributed so that the load is distributed and the scalability is easy. In addition, there is a problem that the availability of data cannot be guaranteed as nodes are dynamically added or deleted, and studies for solving this problem have been actively conducted [6].

2.2. Reinforcement Learning

Reinforcement learning is a method in which an agent defined in a specific environment maximizes a reward among selectable actions based on the current state or a machine learning method. Figure 4 shows the basic process of reinforcement learning, which consists of a set of environmental conditions (S), a set of behaviors (A), and a set of rewards (R). At each point in time t , the agent has a state (S_t) and possible actions (A_t).

Figure 4. Reinforcement learning process.

In addition, an agent takes some action and receives a new state ($S(t+1)$) and reward ($R(t+1)$) from the environment. Based on these interactions, reinforcement learning agents look for policies that maximize cumulative reward. Reinforcement learning is suitable for dealing with the problem of long-term or short-term reward trade-off, and its algorithms include Q-learning, State-Action-Reward-State-Action (SARSA), and SARSA (λ) [7].

In this paper, based on the reinforcement learning model, we find the optimal key sharing parameters (N, T) considering the security and availability of the data. Parameters N and T are used when key shares are distributed to the DHT, each of which means that the total number of key shares and the threshold value needed to obtain the original key K . Based on the key sharing parameters, a reinforcement learning model is applied to obtain as many keys (K) as possible from the DHT network.

In the DHT network environment, state is defined according to the data availability graph. As the structure of the DHT network changes, we define an action that determines key sharing parameters that are different from the existing ones based on the defined state. Based on the similarity of graphs, we find the optimal threshold with the reward value and distribute the key shares to the DHT network.

In particular, an important reason for applying reinforcement learning in the proposed system lies in the dynamically changing characteristics of the DHT.

2.3. Recent Research in the Area of Self-Destructing Scheme

The most basic way to prevent a user's data containing sensitive information from being leaked or to prevent it from being stolen from an attacker is to delete the data manually. However, data passing through the web, such as e-mail, is stored on the server and cannot be controlled directly by the user. As a way to solve this problem, Geambasu proposed a self-destructing scheme that encrypts data and then the key disappears after a certain time. However, the Vanish system using a DHT based P2P network has problems such as Sybil (Hopping) Attack, Sniffing Attack, and time required for encryption and decryption.

Ref. [8] proposed a SafeVanish system that improved the existing Shamir's Secret Sharing algorithm by increasing the complexity of operations required for hopping attack by increasing the range of lengths that each key share can have. In addition, a public key encryption algorithm was used to prevent sniffing attacks. Ref. [9] proposed a SeDas system that utilizes cryptographic techniques and active storage technology based on the T10 OSA standard so that all data and encryption keys are self-destructed or cannot be recovered without user intervention.

Ref. [10] proposed a secure self-destructing scheme for electronic data (SSDD) using existing systems. SSDD first encrypts the data, extracts a portion of the ciphertext, and distributes it to the DHT network along with the encryption key. To obtain plaintext, you must obtain both the encryption key and part of the ciphertext from the DHT network. In terms of security, it has proved its superiority against brute force attack as well as hopping attack and sniffing attack through comparison analysis with an existing system. Ref. [11] proposed a KP-TSABE scheme that utilizes key-policy attribute-based encryption and time-specified attributes for full lifecycle and access control in the process of sharing sensitive data through cloud servers. A cipher text in KP-TSABE can be decrypted only by a user with an attribute whose time attribute is valid and has access to the cipher text. Ref. [12] proposed an IBE-based secure self-destruction (ISS) scheme that cannot only resist against the traditional cryptanalysis and the Sybil attacks but also implement flexible access control. Finally, Ref. [13] proposed an identity-based timed-release encryption (ID-TRE) algorithm and full lifecycle privacy protection scheme for sensitive data (FullPP) using the DHT network.

In addition, several techniques have been proposed to protect privacy without DHT. Tang et al. [14] proposed FADE, which is built on assuredly, deletes files to make them unrecoverable to anyone upon revocations of file access policies. Wang et al. [15] utilized the public key based on a homomorphism authenticator with random mask techniques. However, these techniques depend on specific devices or user interaction, and there are obvious limitations.

Table 1 compares several systems developed based on the previously proposed self-destructing scheme. Unlike the Vanish system, which only guarantees the destruction of the keys required for data encryption, the SSDD and FullPP systems developed later can prevent key encryption, hopping attacks, and sniffing attacks that can be performed in the DHT network. However, the above systems do not mention techniques for ensuring the availability of data for a certain period of time, and thereafter managing key shares whose data must be extinguished. In the system utilizing the self-destructing scheme, the information about the total number of key shares and the threshold value may not be an optimal value over time depending on the environment of the constantly changing DHT network. This is closely related to the basic requirements of the user using the self-destructing system and must be considered. Therefore, in this paper, we propose a technique to set information about key shares using graph similarity and a reinforcement learning model.

Table 1. Comparisons of the security properties.

Security Property	Vanish	Safe Vanish	SSDD	ISS	FullPP
No attacks on VDO before it expires	YES	YES	YES	YES	NO
Whether the key is encrypted or not?	NO	YES	NO	YES	YES
Whether the ciphertext is destructed or not?	NO	NO	YES	YES	YES
Whether the key is destructed or not?	YES	YES	YES	YES	YES
Resistance on the traditional cryptanalysis?	NO	NO	YES	YES	YES
Resistance on the brute-force attack?	NO	NO	YES	YES	YES
Resistance on the hopping attack?	NO	YES	YES	YES	YES
Resistance on the sniffing attack?	NO	YES	NO	YES	YES

2.4. Challenges of Existing Self-Destructing Schemes

The problem of an existing self-destructing scheme is known as a hopping attack using vulnerability of the DHT network and a sniffing attack that can be performed in key share distribution processes. In addition, recent works have been actively conducted to find out whether ciphertexts automatically disappear after a certain time, and how brute-force attacks can be prevented. In addition, it is one of the important problems to be resolved whether the user can specify the desired time independently of the system for the data generated by the self-destructing scheme and freely extend the time. The solution to these problems is mentioned in Section 2.3 of related works.

However, existing studies do not mention a technique for managing key shares in the self-destructing scheme using a DHT network with a problem that cannot guarantee the availability of data due to constantly changing. Several self-destructing systems such as SelfVanish, SSDD and FullPP, which are based on the previous studies, use the fixed parameters proposed in the Vanish system in the process of dividing the key required for data encryption into several pieces. The key sharing parameters (L , C , N , T) mentioned in this paper are the same as those used in the Vanish system, and each has the following meaning. L is a random access key, which indicates location information about key shares, and it is generated as a pseudo random number. C denotes the data encrypted by key K , and N and T , respectively, denote the number of total key shares and the threshold required to obtain the original key K . The Vanish system defines the total number of key shares and the threshold value as $N = 10$, $T = 7$ or $N = 50$, $T = 45$ according to the data. These fixed parameters can largely satisfy the user's requirements, but they are not always optimal values.

Figure 5 shows the availability graph of the data according to different key sharing parameters in the DHT network environment. As can be seen from the measured results, according to the parameters (N , T) set in the self-destructing scheme, data can be accessed before or after the time required by the user (basically 8 h based on VuzeDHT). In addition, even with the same parameters, the results of data availability will be measured differently over time, due to the nature of the dynamically changing DHT network.

Therefore, in order to meet the most important and basic requirements of the self-destructing scheme that enables users to use the data only for a certain period of time, and then to prevent anyone including the user from accessing the data, a study on the technique of setting the number of key shares and the threshold value considering the availability of data must be performed. In this paper, we propose an optimal key sharing parameters setting method to manage key shares based on reinforcement learning algorithm and data graph similarity measurement.

Figure 5. Data availability according to key sharing parameters.

2.5. Mean Absolute Percentage Error

The Mean Absolute Percentage Error (MAPE) is used as a measure of the accuracy of the prediction and is defined as in Equation (1) below:

$$M = \frac{100}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right|. \quad (1)$$

In Equation (1), A_t denotes the actually measured value, and F_t denotes the predicted value. The difference between the actually measured value and the predicted value is divided by the measured value, which is transformed into an absolute value and is calculated as a sum for all the prediction times and divided by the number of points taken. Then, multiply by one hundred and calculate in the form of percentage error.

The basic concept of MAPE is very simple and easy to understand, but there are some drawbacks in practice. Since the division of the measured value is included in the calculation process of MAPE, if A_t is 0, it cannot be applied, and the calculation process of error ratio is not included. Therefore, when MAPE is used to compare the accuracy of predicted results, the predicted value may be set to vary significantly from the actual measured value depending on the given input. The problem of this MAPE can be solved by measuring based on the ratio of the predicted value to the actual value. Generally, to obtain more accurate results, the MAPE of the actual and predicted values is calculated using the geometric mean.

In this paper, we use MAPE to measure the similarity of the data availability graph to key shares in a DHT network environment. Measuring the similarity of data availability graph means how much encrypted data can be decrypted when the corresponding key sharing parameter is applied. In addition, the computation process for measuring the similarity of the graphs should not take much time because the environment of the dynamic DHT network must be monitored and the parameters updated continuously. A_t denotes the availability of an ideal graph to satisfy the requirements of the self-destructing scheme, and F_t denotes the availability graph of the data measured according to the number of total key shares changed by the system and the threshold value.

3. Proposed Mechanism

In this section, the proposed technique is divided into three parts: Section 3.1 describes the overall structure and the process of the proposed system. Section 3.2 describes the similarity measure

of the data availability graph for the reinforcement learning model, and Section 3.3 deals with the reinforcement learning algorithm for the key sharing parameters prediction considering the availability and security of the data.

3.1. Architecture and Procedure of the Proposed System

Figure 6 shows an overview of the overall system using the self-destructing scheme. The proposed system can be installed and operated on all users' equipment using the cloud service, and can be used for cloud storage, email system, instant messages and so on. In addition, the overall system configuration and data flow are not significantly different from those using existing self-destructing schemes. The system installed in the equipment encrypts the user's data and structures it as a new data object based on the additional information required by the system. The additional information needed at this time is mentioned in the self-destructing scheme. The encryption key is divided into several key shares using Shamir's Secret Sharing technique, and each of the key shares are distributed and stored on the nodes in the DHT network. The structure of the data object stored in the cloud storage is in the form of (L, C, N, T) and the decryption process is performed in the reverse order of the encryption process. Naturally, all parameters are needed to decrypt the encapsulated data again.

Figure 6. System architecture.

Figure 7 shows the overall structure of the system using the proposed self-destructing scheme. The system consists of a User Interaction Module for receiving data and access time information from the user, a Cryptography Module for performing encryption of data, and a Key Management Module for managing the keys required for encryption. The Secret Sharing Module for dividing the encryption key into several pieces using Shamir's Secret Sharing, the Threshold Estimation Module for setting the optimal parameters (N, T) based on the similarity measurement between the reinforcement learning and the data availability graph, and a DHT Network Module for distributing the generated key shares to the DHT network.

Figure 7. Structure of proposed self-destructing scheme.

A user who wants to protect sensitive data by using the proposed self-destructing scheme inputs corresponding data through the interface. The system receives an encryption key(K) from the certificate authority (CA) to encrypt the user's data and creates a ciphertext(C). The encryption algorithm uses the Advanced Encryption Standard (AES) algorithm based on a symmetric key algorithm, Cipher Block Chaining (CBC) mode. The key used in the AES-CBC encryption algorithm is 128 bits in size. The Secret Sharing Module divides the encryption key into several shares using Shamir's Secret Sharing technique. We use the number of total key shares(N) and the minimum number of key shares needed to get the original key back(T) as parameters.

In addition, the Threshold Estimation Module periodically measures the availability of data in the DHT network environment and finds the ideal graph and parameters with higher similarity based on the reinforcement learning and similarity of the graph. The generated key shares are distributed to the DHT network by the DHT Network Module.

3.2. Measuring Similarity of the Data Availability Graph

In the proposed data self-destruction scheme, data availability is calculated by obtaining the key K for decrypting the encrypted data. In this process, MAPE is used as a method to measure the similarity with the ideal state (all data is available during the user-specified time). For example, a user divides 10 encryption keys K into 10 key shares and distributes a total of 100 key shares to the DHT. If the user gets five encryption keys K based on the key shares from DHT at a particular time, then the data availability is 0.5.

As mentioned in Figure 5, in the self-destructing scheme, the availability graph of the data is measured differently according to the number of the key shares and the threshold value. Therefore, in this paper, we compare the availability graph of ideal data with the availability graph of measured data in actual DHT network environment using Mean Absolute Percentage.

Figure 8 shows how to measure the similarity to the ideal graph, which guarantees the use of data only for a user-specified time. The proposed scheme uses MAPE, which is calculated by adding the difference of the result value according to the input value in the similarity measurement process. The result value (y -axis) along the input value (x -axis) is below the ideal graph, meaning that key shares are self-destructed before the user-specified time, which means that data availability is low. On the other hand, if the result value is above the ideal graph, meaning that key shares are not self-destructed and remain in the DHT network. This may cause system performance degradation or privacy problems. We find optimal key sharing parameters that are similar to the ideal graph by using MAPE, and apply them to the data self-destruction scheme.

Figure 8. Measuring similarity of data availability graph.

The following is an example of an ideal graph and similarity measure using MAPE for the availability graph of data according to different threshold values in VuzeDHT environment. In this

case, the total number of key shares is 10, and the threshold values are 40 and 90, respectively. The data availability graph divides the keys required to encrypt each of the 100 data pieces and distributes them to the DHT network. After 12 h, each key shares are checked to see if the original encryption key can be recovered or not.

Table 2 is an example of measuring the similarity of the graph using the data of Figure 8. The total number of key shares(N) is equal to 10, and is the result of the MAPE value and similarity when the minimum number of key shares(T) to obtain the original encryption key is 4 and 9, respectively. In case of $T = 9$, the similarity with the ideal graph was found to be lower than in the case of $T = 4$. In the above example, the total number of values used in the MAPE calculation is 120 (1 in 6 min, 10 in 1 h), and the more the values are used, the higher the accuracy of the calculation. However, due to the nature of the DHT network, which is heavily distributed all over the world, there is a clear limit to minimize the time required for the lookup operation. In addition, if the availability graph of the data is measured before the lookup operation is completed, a situation may arise in which the key fragment does not receive an accurate result even though it exists in the DHT node. Therefore, an operation that takes a value for measuring the similarity of a graph must be performed after the LOOKUP operation is definitely finished.

Table 2. MAPE based similarity of graph.

Threshold (%)	MAPE	Similarity of Graph
T = 40	23.72	76.28
T = 90	35.68	64.32

3.3. Reinforcement Learning Algorithm for Key Sharing Parameters

We apply reinforcement learning as a method for setting optimal parameters for Shamir's Secret Sharing considering the availability and security of data in the self-destructing scheme environment. The characteristics of reinforcement learning are expected to be better by defining the state and choosing the action that can obtain the best result at a certain point in time. In this process, we update the reward that each state and action can have. A state-action table (Q-table) is updated until a target value of the same value is finally found for a state and an action.

In the self-destructing scheme proposed in this paper, the optimal threshold is found based on the security and availability of data based on the reinforcement learning model. We define the state according to the data availability graph in the environment of the DHT network, and, based on the defined state as the structure of DHT network changes, the similarity of graph according to the action that determines the number of key shares and threshold value for finding the optimal key sharing parameters with the reward and distribute the key shares to the DHT network. Due to the continuously changing characteristic of DHT, data availability of the same key sharing parameter is not kept constant. Therefore, the similarity measuring process with the ideal state should be done very quickly. MAPE is a simple method for comparing the result according to the input value, and the key sharing parameters having high similarity, which is found through the iterative process and applied to the system.

The state for applying reinforcement learning has data availability in the form of a matrix over time for a specific parameter(N, T). This includes the availability of data at each time in units of time intervals to measure the similarity of the graph. Action is the process of selecting specific parameters defined in State. This process requires a proper balance between exploration applying new values and exploitation applying existing values.

The reason for choosing reinforcement learning to set the key sharing parameters required for the self-destructing scheme is that the optimal values to be applied at each point in time vary depending on the dynamic change of the DHT network environment. In addition, since the initial default parameters are expected to yield results similar to the final goal, the parameters defined by the state and action sets are in the form of normal distributions for the default values.

Table 3 shows the parameters for applying reinforcement learning. In order to apply reinforcement learning algorithm, state, action, and reward must be defined basically. State is defined as the availability of data on the number of key shares() and the time that key shares generated by the threshold() are distributed to the DHT network.

Table 3. Reinforcement learning parameters.

State	Key Sharing Parameters(N, T)
Action	Select and apply the parameters
Reward	Similarity with ideal state

State stores the availability value of data for a specific time unit in the form of a simple one-dimensional array. It is in the nature of the data availability graph that state is defined as a discrete array rather than a continuous form of expression. The data availability graph in the DHT network environment is independent from the measured values in relation to the measured time units.

Therefore, it is very inefficient to define and apply the formula for each measurement time unit. In addition, it is much more efficient to store only the measured values without defining the state as a formula because it requires only the measurement values according to time unit in the process of measuring the similarity of the graph using MAPE. However, if the state is defined as a discrete form, only the measured values for the fixed time interval can be applied.

Action is a process of selecting the maximum reward value among the defined states. There are exploration methods for applying the new value to the result and exploitation for applying the existing value as it is. The method of selecting two methods is to apply decaying ϵ -greedy to perform exploration at a high probability in the initial stage and increase the probability of performing exploitation over time.

In the Self-Destructing Scheme environment, the smaller the time unit of measuring the availability graph of data, the more accurate the result, but it can be a disadvantage in terms of calculation time. This is also related to the time required for the lookup operation to query the data in the DHT network. In this paper, we measure the availability of data 10 times in 1 h (once every 6 min):

$$Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]. \quad (2)$$

The process for updating the Q-table constructed at the time based on the defined state and action is defined as shown in Equation (2). $Q(s_t, a_t)$ returns the probability for the next state based on state and action in time. α means learning rate, and, as this value increases, existing information is updated with new information. This is related to learning rate and has a value between 0 and 1. This means reward that is a compensation as it changes from a specific time. Moreover, it determines how much the reward at the discount rate is worth the future reward. This can be reflected in lesser and lesser rewards expected to come in the future:

$$\text{Reward}(t) = 1 - \frac{1}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right|. \quad (3)$$

Reward is defined based on MAPE at the time and converted to the appropriate form for applying to the Q-table. In order to meet the requirements of the self-destructing scheme, the difference between the ideal graph and the total number of key shares and the data availability graph measured by the threshold is calculated by the MAPE as a percentage, so the result subtracted from 100 is the degree of similarity between two graphs. In addition, convert the result calculated as a percentage to a decimal point to apply to the Q-table.

The initial Q-table for applying the reinforcement learning algorithm is defined as $Q_1(s,a)$ with default values of $N = 10$, $T = 7$ and $Q_2(s,a)$ with $N = 50$, $T = 45$ as default values. Each Q-table should not be duplicated in the normal distribution form based on the default value.

Figure 9 shows the process for setting the total number of key shares (N) and threshold (T). First, key fragments are generated and distributed to the DHT network. In addition, it measures the availability graph of user's data in the DHT network environment. After that, we check the similarity with the ideal data availability graph, which is the target value finally, and select another parameter (N , T) in the process of generating the key shares. Through this iterative process, the degree of similarity with the ideal graph, that is, the reward being large, will be selected.

Figure 9. Key sharing parameter estimation procedure.

This iterative process ends when the data availability graph according to the parameters predicted through reinforcement learning matches perfectly with the ideal data availability graph. However, due to the nature of the ever-changing DHT network, it is virtually impossible to perfectly match the availability graphs of two different data. Therefore, while the self-destructing scheme is running, the above iterative process will be performed continuously in the background.

Algorithm 1 represents our proposed solution for finding optimal parameters by applying reinforcement learning in a system of self-destructing scheme. The initial $Q(s,a)$ value is generated to have an arbitrary normal distribution without duplication based on the values $N = 10$, $T = 7$ or $N = 50$, $T = 45$. In the reinforcement learning process, the data availability graph for the corresponding state and action is measured to obtain the reward. The reason for having a normal distribution here is that it assumes that the key shares are generated with a default value in the system utilizing the existing self-destructing scheme and have relatively high availability and security. In addition, the weights are set differently for each variable, so that the deviation of the threshold values is set to be larger than the total number of key pieces.

Algorithm 1. Reinforcement Learning

Algorithm: Reinforcement Learning based Threshold Estimation

Initialize: $Q(s, a)$ arbitrarily

Repeat (for each episode)

Initialize s

Chose a from s using Q (ϵ -greedy)

Repeat (for each step of episode)

Take action a , observe r, s'

Choose a' from s' using Q (ϵ -greedy)

$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma Q(s', a') - Q(s, a)]$

$s \leftarrow s'; a \leftarrow a'$

Until s is terminal

4. Evaluation

Table 4 shows the environment for performance evaluation. A Peersim-1.0.5 simulator is used to construct an environment for experimenting DHT networks. The DHT protocol used VuzeDHT with a fixed timeout of 8 h. The total number of nodes constituting the DHT network is 1000, and the number of malicious nodes is 50, which is 5% of the total. The malicious node means the abnormal node that does not transmit data to other nodes in the DHT, and when the key shares are transmitted to the malicious node, the key shares cannot be restored. If a key share generated through the self-destructing scheme is distributed to a malicious node in the DHT network, the key share cannot be retrieved from the node and is taken by the attacker.

To the best of our knowledge, finding optimal key sharing parameters in data self-destruction scheme has never been conducted by the research community. We have configured the same environment as the existing system to ensure the reliability of the experimental results. The whole experimental environment is the same as the Vanish system, and the parameters for generating key shares are the same. The total number of key shares and thresholds applied to the system are set to have the values of $N = 10$, $T = 7$. To ensure higher security, the values are set to $N = 50$, $T = 45$. In addition, AES (Advanced Encryption Standard) algorithm based on CBC (Cipher Block Chaining) mode, which is widely used as a symmetric key algorithm, is used as the encryption algorithm applied to the system. The key size used for data encryption is 128 bits, and the key shares generated by Shamir's Secret Sharing process are each 16 to 51 bytes in size.

Table 4. Parameters for evaluation environment.

Simulator	Peersim-1.0.5
DHT Protocol	VuzeDHT
Number of DHT Nodes	1000
Cryptography	AES-CBC
Length of Key	128 bits

Table 5 shows the amount of data used in the environment for performance evaluation, the parameters for generating key fragments, and the measurement time. For the experiment, 1000 pieces of data were used, and each piece of data was encapsulated with a different encryption key. The key sharing parameters for generating key shares were set to have a normal distribution with a value based on the default parameters and high security parameters used in the Vanish system. Because the VuzeDHT network has a fixed timeout of 8 h, the data availability graph is measured for 12 h for each amount of data to get accurate results.

Table 5. Evaluation scenario.

Number of Used Data	1000
Key Sharing Parameters (default)	$N = 10$, $T = 7$
Key Sharing Parameters (high security)	$N = 50$, $T = 45$
Measuring Time (h)	12

In the first experiment, the key sharing parameter is applied to the Vanish system and the proposed system to compare the similarity with the ideal graph when measuring the data availability graph. The process of applying reinforcement learning should take at least 12 h. Therefore, state, action, and reward updates are applied to the 100 data that has been measured in the availability graph of the data.

In the second experiment, the availability graph of the data according to the default and high security parameters is measured while the learning process for the given data is completed. Each data availability graph has a larger reward value as the degree of similarity with the ideal graph is higher.

4.1. Similarity with Ideal Graph

Figure 10 compares the similarity between the Vanish system and the proposed system with the ideal graph when the data availability graph is measured by applying the key sharing parameter. The update period of state, action, and reward is set to the point when the availability graph measurement for 100 data is completed. The above experiment is the result of measuring the highest similarity to the ideal graph among the applied key sharing parameters when all the data of 100, 500, and 1000 data are reflected.

Figure 10. Comparing the similarity of availability graph.

In the initial stage of the experiment, the data availability graph of the Vanish system using fixed key sharing parameters was measured to be more similar to the ideal graph. On the other hand, as the learning process using the reinforcement learning model is repeated, the similarity with the ideal graph is higher than the existing scheme. This means that the proposed scheme can better assure the availability and security of the data. If the learning process is not repeated enough, the user's data may not be accessible before the specified time. In this respect, the proposed self-destruction scheme has limitations. This is because the proposed system uses new key sharing parameters in the process of applying the reinforcement learning model, and these limitations can be solved easily by separating agents for initial learning. Separating the agent for learning and applying optimal key sharing parameters after sufficient iterations can be a good extension of our work. However, as the amount of data used for learning increased, the proposed system could guarantee the availability and security of the data.

4.2. Comparative Analysis of Different Learning Models

Figure 11 illustrates the similarity measure with the ideal graph according to three different learning models, namely regression analysis, q-learning and SARSA. Each learning model applies the default key sharing parameter and iterates 1000 times, and all other experimental environments are the same. The regression analysis model is implemented by selecting the next input value based on the result (similarity) of the key sharing parameters. Q-learning and SARSA models select the next key sharing parameters based on the predefined state, action and reward.

Figure 11. Comparing similarity according to learning model.

The results of the regression analysis were not significantly increased or decreased as the learning process was repeated. Q-learning and SARSA models increase in similarity to the ideal graph as the learning process is repeated. However, SARSA was found to have a higher similarity than q-learning because sufficient exploration was ensured using the following input values in the learning process.

4.3. Availability Graph with Default Parameters

Default key sharing parameters are used when the user wants to guarantee the privacy of data (not explicitly specify the high security parameter) by using a self-destruction scheme. It generates a smaller number of key shares than the high security parameter, so it takes less time to encrypt/decrypt the data. Figure 12 shows the availability graph of the data according to the key sharing parameter set to default when the reinforcement learning process for the given data is completed. The data availability graph for each parameter has a higher reward value as the degree of similarity with the ideal graph is higher.

Figure 12. Availability graph with default parameters.

Analysis of the experimental results show that the highest similarity to the ideal graph was obtained when $N = 11$, $T = 7$. In addition, when the graph of data availability is measured according to key sharing parameters with $N = 10$, $T = 7$ and $N = 9$, $T = 6$, it is measured that the similarity with an ideal graph is relatively high.

4.4. Availability Graph with High Security Parameters

Figure 13 shows the results of applying the key sharing parameter with high security and the same experiment as Figure 11. Likewise, when the results of the experiment are analyzed, the similarity with the ideal graph is the highest when $N = 52$, $T = 40$.

In contrast to the experimental results in Figure 12, the availability graphs of the measured data in Figure 13 shows that the similarity with the graph is relatively low. This is because the initial threshold value is different from the default parameter and the high security parameter. In general, high security parameters are used to ensure more confidential data privacy. Since the high security parameter has a strict threshold value, data availability is therefore lower than the default parameter.

Figure 13. Availability graph with high security parameters.

5. Conclusions

In this paper, we propose a technique for setting key sharing parameters based on a reinforcement learning model that is using similarity of graphs to satisfy a higher level of data availability and security in the self-destructing scheme. The reinforcement learning model applied in this paper defines the state according to the data availability graph in the DHT network environment and action to select the key sharing parameter based on the defined state as the DHT network structure changes. In addition, the similarity of the graph is found by reward to find the optimal threshold value. MAPE is used as a method to measure the similarity of the graph. In order to satisfy the requirements of the self-destructing scheme, the graph of the availability of data measured according to the key sharing parameters modified by the system is compared with a predefined ideal graph.

To evaluate the performance of the proposed self-destructing scheme, we used a similar experimental environment as used by the Vanish system. As a result, both availability and security were guaranteed in the constantly changing DHT network environment. Therefore, if the key sharing parameter is set by applying the proposed technique, it is expected that the user can access the data only for the desired time in the self-destructing scheme environment, and then automatically disappear after that.

The process of the reinforcement learning model that sets key sharing parameters in consideration of both security and availability can only be achieved by obtaining the results of the availability graph of the data. Of course, the time it takes depends on the DHT network environment, but can be optimized using parallel processing or clustering techniques. We will also conduct research to set the time, in minutes or hours that allows users to access data using various DHT network protocols. Based on the proposed method of applying optimal key sharing parameters, we hope that the data self-destruction scheme will be applied in real life such as instant message, cloud storage and e-mail server.

Author Contributions: Y.K.K. led the research of data self-destruction for secured data. S.U. worked closely together with Y.K.K. and was involved in every section of this manuscript. All the research work is carried under the supervision of professor C.S.H. The data used in the experiments contains sensitive medical information and must be transmitted securely. K.K., Y.J. and J.L. provided their expertise in acquiring, cleaning and analyzing the data and helped in carrying out the experiments. However, all the authors contributed equally in finalizing this manuscript.

Acknowledgments: This work was partly supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. 2017-0-00735, The S/W development to extract, refine and duplicate a real-time data from various database) and Korea Evaluation Institute of Industrial Technology (KEIT) grant funded by the Korea government (MOTIE) (No. 2017-0-00735, The S/W development to extract, refine and duplicate a real-time data from various database). Choong Seon Hong is the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Verma, A.; Kaushal, S. Cloud computing security issues and challenges: A survey. In Proceedings of the International Conference on Advances in Computing and Communications, Kochi, India, 22–24 July 2011.
2. Geambasu, R.; Kohno, T.; Levy, A.A.; Levy, H.M. Vanish: Increasing data privacy with self-destructing data. In Proceedings of the 18th Conference on USENIX Security Symposium, Montreal, QC, Canada, 10–14 August 2009.
3. Haeberlen, A.; Kouznetsov, P.; Druschel, P. PeerReview: Practical accountability for distributed systems. In Proceedings of the 21st ACM SIGOPS Symposium on Operating Systems Principles, Washington, DC, USA, 14–17 October 2007.
4. Shamir, A. How to share a secret. *Commun. ACM* **2016**, *22*, 612–613. [[CrossRef](#)]
5. Xie, M. P2P Systems Based on Distributed Hash Table. Master's Thesis, Department of Computer Science, University of Ottawa, Ottawa, ON, Canada, 2003.
6. Rhea, S.; Geels, D.; Roscoe, T.; Kubiatowicz, J. Handling churn in a DHT. In Proceedings of the USENIX Annual Technical Conference, Boston, MA, USA, 27 June–2 July 2004.
7. Kaelbling, L.P.; Littman, M.L.; Moore, A.W. Reinforcement learning: A survey. *J. Artif. Intell. Res.* **1996**, *4*, 237–285.
8. Zeng, L.; Shi, Z.; Xu, S.; Feng, D. SafeVanish: An improved data self-destruction for protecting data privacy. In Proceedings of the IEEE International Conference on Cloud Computing Technology and Science, Indianapolis, IN, USA, 30 November–3 December 2010.
9. Zeng, L.; Chen, S.; Wei, Q.; Feng, D. SeDas: A self-destructing data system based on active storage framework. In Proceedings of the Asia-Pacific Magnetic Recording Conference, Singapore, 31 October–2 November 2012.
10. Wang, G.; Yue, F.; Liu, Q. A secure self-destructing scheme for electronic data. *J. Comput. Syst. Sci.* **2013**, *79*, 279–290. [[CrossRef](#)]
11. Xiong, J.; Liu, X.; Yao, Z.; Ma, J.; Li, Q.; Geng, K.; Chen, P.S. A secure data self-destructing scheme in cloud computing. *IEEE Trans. Cloud Comput.* **2014**, *2*, 448–458. [[CrossRef](#)]
12. Xiong, J.; Yao, Z.; Ma, J.; Li, F.; Liu, X. A secure self-destruction scheme with IBE for the internet content privacy. *Chin. J. Comput.* **2014**, *37*, 139–150.
13. Xiong, J.; Li, F.; Ma, J.; Liu, X.; Yao, Z.; Chen, P.S. A full lifecycle privacy protection scheme for sensitive data in cloud computing. *Peer Peer Netw. Appl.* **2015**, *8*, 1025–1037. [[CrossRef](#)]
14. Tang, Y.; Lee, P.P.C.; Lui, J.C.S.; Perlman, R. FADE: Secure overlay cloud storage with file assured deletion. In Proceedings of the International Conference on Security and Privacy in Communications Systems, Singapore, 7–10 September 2010.
15. Wang, C.; Wang, Q.; Ren, K.; Lou, W. Privacy preserving public auditing for storage security in cloud computing. In Proceedings of the IEEE Conference on Computer Communications, San Diego, CA, USA, 15–19 March 2010.

© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).