2015 IEEE 29th International Conference on Advanced Information Networking and Applications

Content access control scheme for P2P networks using a reputation value

Kentaro Aburada
Oita National College of Technology
Oita, 870-0152, Japan
Email: aburada@oita-ct.ac.jp

Yoshihiro Kita
and Mirang Park
Kanagawa Institute of Technology
1030 Shimo-ogino, Atsugi,
Kanagawa, 243-0292, Japan

Naonobu Okazaki
University of Miyazaki
Miyazaki, 889-2192, Japan

***Abstract*—In recent years, with the improvement of the high speed communication infrastructure, P2P content distribution systems have been attracting more attention. In a P2P content distribution system, the lack of a central management server provides the system its robustness. However, it also leads to problems in content reliability and accessibility. We propose a secure content distribution system with improved accessibility by introducing a secret sharing scheme.**

I. INTRODUCTION

In recent years, peer-to-peer (P2P) content distribution systems have been attracting attention because of improvements in the high speed communication infrastructure. In a P2P system, the lack of a central management server results in a system which has fault tolerance and scalability. For example, client–server systems approach a performance limit as the number of client nodes is increased . On the other hand, a P2P system can continue to perform smoothly as nodes are added, provided that communication bandwidth is sufficient. However, a P2P content distribution system, because of the lack of a central management server, has problems in terms of the reliability of the distributed content. For example, a malicious node can easily distribute fake content or content containing a virus program.

Digital signature schemes exist to assure the integrity of content. Such schemes can check who generated the content and whether data has been falsified. However, digital signature schemes require that a certificate authority be set up. Therefore, if a digital signature scheme is used directly in a P2P network, this reduces its decentralized nature.

As an alternative, Palomar have proposed a certification scheme which uses multiple digital signatures[1]. This scheme generates content certificate that uses multiple signature, and generates access certificate to content. This scheme is able to obtain protection of content and access control without reducing the advantages of P2P. However, it is difficult to select a reliable node to handle the multiple signatures in a P2P environment. Specifically, the problem of not being able to access content as long as the owner is disconnected from system exists. Therefore, it is impractical in light of the P2P environment to have a high incidence of entry and exit of nodes.

In this paper, we propose a P2P content distribution system which is intended to secure the distribution of content. This system implements a trust management system and obtains key sharing with multiple nodes that use a secret sharing scheme. In this system, a trust management system generates criteria of selection of reliable nodes that multiple signatures of contents certificate in a P2P environment, and distributes the information of the key, and shares key using multiple nodes to assure the access available of content.

We describe method previously proposed by Palomar and others in Section 2. Our proposed system is described in Section 3. We describe the evaluation and consideration our system in Section 4. Section 5 concludes the paper with a discussion of future work.

II. RELATED WORKS

In this section, firstly, we describe an access control scheme which generates a content certificate. Secondly, we describe a secret sharing scheme, a (k,n) threshold scheme, and a trust management system for the problem of interest.

A. Multi-signature-based access control scheme

Palomar's method of implementing an access control scheme generates a content certificate based on a multi-signature to obtain protection of content and access control that depends on the collaboration of only a few nodes[1]. This method consists of the following three subprotocols.

1) Join subprotocol
2) Content authentication subprotocol
3) Content access subprotocol

Content is encrypted according to security labels, such as the classic set "confidential", "restricted", "secret", and "top-secret". Firstly, in the Content authentication subprotocol, the content owner selects reliable nodes and generates a content certificate that depends on a multi-signature created by the selected reliable nodes. Secondly, in the Join subprotocol, the content owner generates an access certificate for a request node. This access certificate encloses the information to decrypt an encrypted content decryption key. Finally, in the Content access subprotocol, the request node contacts the content owner and gets the decryption key. Hereby, the request node can access the content. A node without an authorization

1550-445X/15 $31.00 © 2015 IEEE
DOI 10.1109/AINA.2015.232

certificate can decrypt content only by a brute force attack. In reality, an attacker cannot access the content because this task is beyond the capability of the resources of an attacker. However, with this scheme, the problem remains of how to set the criteria of the selection of a reliable node among the anonymity of a P2P environment. And, if the owner exited the system, content accessibility becomes a problem because the decryption key cannot be obtained. In this paper, we discuss how to achieve a secure content distribution system with accessibility by the introduction of a trust management system and secret sharing scheme.

B. (k,n) threshold scheme

The (k,n) threshold scheme is a secret sharing scheme proposed independently by Blakley[2] and Shamir[3]. In this scheme, secret information is divided into n pieces in such a way that it is easily reconstructible from any k pieces, but is not reconstructible from any $k-1$ pieces. In addition, this scheme has the advantage that the secret information as cannot be easily guessed.

C. Trust management system

A P2P content distribution system requires a high reliability to avoid malicious content. Schemes for rating node reliability include the "peer-profile based"[4], "recommendation-based"[5] and "reputation-based"[6], [7], [8], [9], [10], [11], [12], [13], [14], [15] methods. The reputation-based trust management system is best for constructing a trust model in a P2P environment because it creates a rating based on reputations reported by multiple sources. In this paper, we implement a reputation-based trust management system which evaluates reputation according to past transactions[6].

III. Proposed system

In this section, we give an overview of our system and describe in detail each subprotocol.

A. System overview

We propose a secure content distribution system with accessibility in a P2P network. Our system uses a trust management system for generating reputation.

This system consists of four subprotocols.

1) Join subprotocol
2) Decryption key distribution subprotocol
3) Content authentication subprotocol
4) Content access subprotocol

(i) The content owner selects reliable nodes which then collaborate to generate a content certificate based on reputation. The owner generates the content certificate according to a multi-signature created by the selected reliable nodes. Then, (ii) content is encrypted using a common-key cryptosystem. The decryption key is distributed on the network according to a secret sharing scheme. (iii) A request node requests an access certificate from the content owner. The owner decides whether to generate an access certificate; a generated access certificate includes information on the decryption key. (iv) The request node obtains a decryption key using the access certificate and accesses content. Normally, a user which has an access certificate can access content even when the content owner is disconnected from the system.

After describing the trust management system in detail, we will describe each subprotocol.

	Requester	Owner
Increase	Content transaction is success	Content transaction is success
Decrease	Content transaction is failure Content is abnormal	–

TABLE I
Updating of a reputation value.

B. Trust management system

In this section, we describe a trust management system which evaluates a reputation based on past transactions. We define updating criteria of reputation and a trust vector for use in reputation generation.

1) Updating criteria reputation: Reputation rises and falls according to past actions. If content is distributed, it is important that content transactions take place normally. Frequent failure of content transactions has an significant impact on content distribution. If a node receives malicious content in the form of falsified or virus-infected content, the request node decreases the reputation of the providing node. Then, if the certificate was normal when the request node received the malicious content, the request node decreases the reputations of nodes which disclosed content or collaborated to generate the certificate. On the basis of the above-mentioned reputation update method, a trust relationship can be determined for normally distributed content. The updating criteria of reputation are shown in Table I.

2) Trust Vector: Trust vectors are binary vectors of single-bit values corresponding to past transactions[6]. A trust vector is updated according to the result of a download or query request and is stored along with the number of significant bits. The stored bit is 1 following an honest transaction or 0 following a dishonest transaction, as shown Table 1. Figure 1 shows the updating process if reputation increases following a request node N conducting a honest transaction with owner O. Each node stores a trust vector along with a node ID and a significant bit. For example, node N stores the trust vectors of nodes A, B, and C, which have node IDs u_A, u_B, and u_C, as shown in Figure 2. Reputation as a trust rating and distrust rating is based on the expression below, where the trust vector is v, the 1's complement of the trust vector is w, and the number of significant bits is m.

$$\begin{aligned} \text{Trust rating} &= \frac{v}{2^m} \\ \text{Distrust rating} &= \frac{w}{2^m} \end{aligned} \tag{1}$$

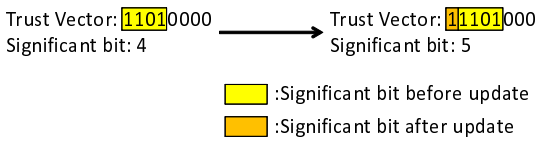

Fig. 1. Updating process of the trust vector.

node ID	Trust vector	significant bit
u_A	11101000	5
u_B	10100000	3
u_C	11010111	8

TABLE II
TRUST VECTOR TABLE.

3) Reputation value acquisition process: Node A broadcasts a trust query to every reliable node when it wants to know the reputation of node T. If node I receives the query, it replies with the reputation of the relevant node. Then, node A calculates a weighted reputation rep_1, rep_2, ... , rep_n which consists of the received value multiplied by the reputation of each node i stored by node A. If the relevant node was unknown to the receiving node, the latter sends a query recursively to every reliable node. Requesting node A obtains reputations of nodes through repetition of the above-described process. Then, node A sorts the reputations in descending order and takes the data of k reputations and takes a simple arithmetic average.

$$\frac{\sum_{i=1}^{k} rep_i}{k} \tag{2}$$

Node A constructs a rating of node T on the basis of the averaged reputation. The above-described reputation value acquisition process is shown in Figure 2. The proposed system makes a decision regarding selecting reliable nodes and passing the decryption key according to the resulting reputations.

C. Subprotocols

In this section, we describe in detail each subprotocol. Every node has two pairs pairing up a private key with a public key which validate each secret communication and generation of an access certificate. We show the items used in this paper as below.

- u_i: node ID of node i
- m: content
- x: data
- $H(x)$: hash value of data x
- PK_i: public key of node i
- SK_i: private key of node i
- VPK_i: public key which validates the access certificate of node i
- VSK_i: private key which generates the access certificate of node i
- K_m: common key which encrypts and decrypts content m

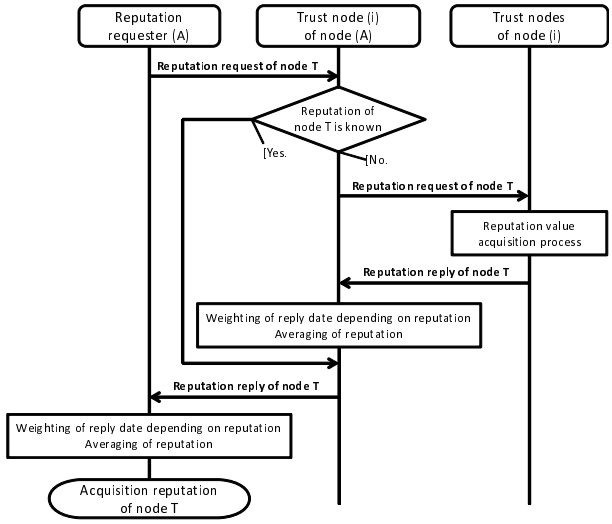

Fig. 2. Reputation value acquisition process.

- $enc_K(x)$: encryption of data x which uses key K
- $S_i(x) = enc_{SK_i}(H(x))$: Signature of Node i for data x
- C_B^A: certificate generated by A for B
- $Share_k$: share of ID k
- OLS: list of trusted nodes

1) Join subprotocol: The join subprotocol consists of these processes: selection of a reliable node, distribution of the signature and generation of the content certificate, and encryption content. The owner runs these processes when it discloses content.

We describe below in detail the Join subprotocol.

(1) Owner O selects n nodes from reliable group and constructs list $OLS =< u_O,\ u_1,\ u_2,\ldots,\ u_n >$ for generating a content certificate. Reliable nodes are selected in descending order of reputation as determined by the owner's reputation table. (2) The owner generates content certificate $C_0 =< C,\ S_O(C) >$ and the first signature used in the multi-signature. (3) The owner begins the distribution signature process and sends C_0 with content m the next node in list OLS, node 1. (4) Node 1 validates content m and signature C. Validation of content m consists of making a comparison between the generated hash value of the received content m and hash $H(m)$ in certificate C. If the hash values are the same, node 1 allows accurate identifiable that signature applying content is identical with validated content. Therefore, our system can prevent the falsification of a content certificate. In addition, the system determines if content contains a malicious program by means of existing technology, such as virus check software. (5) If neither of these two problems exist, node 1 makes a signature and $C_1 =< C_0,\ S_1(C_0) >$ and sends content m and C_1 to node 2. Then, node 1 sends a message to owner O and updates the list of the signed certificate portion. If a problem did exist with respect to validity, the node stops the multi-signature process and sends an abnormal exit

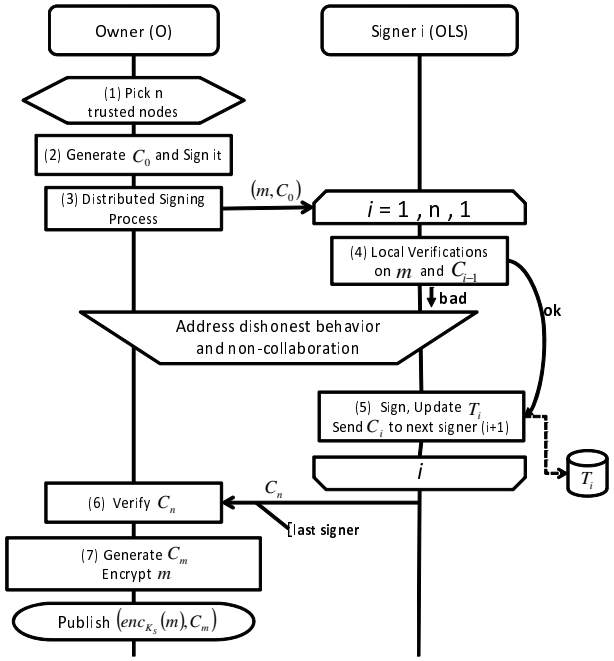

Fig. 3. Content certification issue process.

message to the owner. Therefore, if node i receives a signature and content, node i validates the content and generates the signature $C_i =< C_{i-1},\ S_i(C_{i-1}) >$. Then, node i sends content m and signature C_i to node $i+1$. The above process repeats until the final node of OLS is reached. If the signature process reaches node n, the final element of OLS, then node n sends multi-signature C_n to the owner. (6) If the owner receives multi-signature C_n, the owner validates it. (7) The owner discloses the content certificate as $C_m =< C,\ C_n >$ and encrypted content $enc_{K_m}(m)$ by means of a common key cryptosystem.

2) Decryption key distribution subprotocol: This subprotocol distributes a decryption key of content, K_m, depending on a secret sharing scheme and shares key information over the network. Firstly, the owner distributes the common key K_m which is used in encryption and decryption of content according to the (k,n) threshold scheme and creates share of n. The owner calculates the node ID of the destination of each share, $regID$, using the following expression with random digits R.

$$regID_j = H(ID_m \oplus R \oplus j)\ (j = 1, ..., n) \tag{3}$$

Where $\oplus$ is character concatenation.

Owner sends $message(j)$ consisting of the share, random digits R, number of distribution n, threshold k, public key for validating access certificate VPK_O, and threshold of reputation λ to the calculated node i by means of secret communication using a public key cryptosystem.

$$message(j) =< Share_j, R, n, k, VPK_O, \lambda > \tag{4}$$

Our system can prevent access failure of content due to a DoS attack by hiding the destination of share from malicious nodes.

3) Content authentication subprotocol: We next describe in detail the processes of the content authentication subprotocol. Firstly, node i calculates its node ID by the below expression when it desires specific content.

$$u_i = H(IPaddress \oplus port) \tag{5}$$

Then, (1) node i requests the generation of an access certificate from the owner in order to access specific content. It runs this process using two-way authentication with a public key cryptosystem in order to prevent the generation of a malicious access certification by spoofing. Node i generates a request for information $req =< u_O,\ u_i,\ RF >$ consisting of the source node ID, destination node ID, and generation request of an access certificate from owner O. Node i encrypts req using its private key SK_i and sends plain text and encrypted text of the request information to the owner. (2) Owner O gets the public key of request node i and validates the received request. If the received request has no problems, the owner can identify with certainty that the received request information as from node i. After validation, the owner collects the reputation of request node i and calculates whether to generate an access certificate according to the reputation. (3) If the owner O generates a certificate based on its calculation, the owner makes certificate $C_i =< u_O,\ u_i,\ t >$. Here, t is the expiration date of the certificate. The owner encrypts the certificate using its private key SK_O and sends plain text and encrypted text as a reply to node i. (4) After node i has receives the information from owner O, node i gets the public key of the owner PK_O and validates the access certificate. Node i can verify the authenticity of the access certificate received from the owner. If the access certificate has no validation problems, node i encrypts C_i using its private key SK_i and sends plain text and encrypted text of C_i as a confirmation message to the owner. (5) If the received confirmation message has no validation problems, owner O encrypts C_i using its private key for generating certificates VSK_O. Encrypted information $enc_{VSK_O}(C_i))$ is access certificate $CERT_i^O$. The owner makes the access certificate $C_i^O =< CERT_i^O,\ R,\ t >$ by adding to $CERT_i^O$ the expiration date t and random digits R by using distribution of decryption key. Then, the owner encrypts C_i^O by using the public key of request node i and sends certificate message $mes =< enc_{PK_i}(C_i^O),\ S_O(enc_{PK_i}(C_i^O)) >$ to the request node. (6) After sending, the owner O updates the list of the generation certificate. (7) If the signature does not have validation problems according to the received message, request node i stores C_i^O as a normal access certificate.

4) Content access subprotocol: Content access subprotocol consists of five processes: confirmation of the access certificate, search for a share node, validation of giving of shares, collection share of required number, and getting the content decryption key.

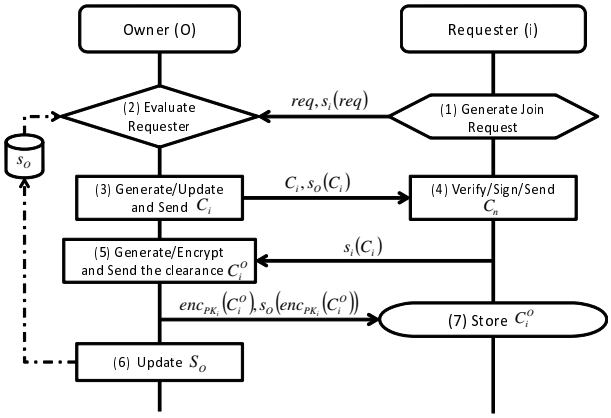

Fig. 4. Permission certificate issue process.

We describe in detail the processes of the content access subprotocol. For a node to access content, it needs an access certificate for the specific content. (1) Therefore, node i checks whether it has an access certificate. If node i does not have an access certificate, it requests access certificate generation from owner O. If node i has an access certificate, node i checks whether it has a valid certificate. If node i has an expired certificate, it requests access certificate generation from the owner in the same way as in the case of not having an access certificate. (2) If request node i has a valid access certificate, it calculates the share destination node ID $regID_j$ by expression 3 using random digits R written in certificate. Node i obtains the list of sharers by calculating n times written in certificate. Then, node i sends a share request message and certificate $CERT_i^O = enc_{VSK_O}(C_i)$ in an access certificate to the calculated sharers. Here, data is sent encrypted using the public key of the source node. (3) Receiving sharer node j validates certificate $CERT_i^O$ by using its key VPK_O. Node j decrypts the received certificate by using its key VPK_O and checks C_i. (4) Sharer node j evaluates the reputation of request node i. If the reputation satisfies the threshold λ, sharer node j gives share. Here, when the sharer gives share, it uses secret communication using a public key cryptosystem. If this process develops problems midstream, sharer node j finishes the process of delivery and receipt and sends a finish message. (5) If request node i has an access certificate and a high reputation, it can collect the share k needed for decryption and obtains the decryption key for the content and can then access the content.

IV. Evaluation

A. Simulation environment

Firstly, we evaluate the integrity of content, which can get normal content when nodes requested content. Accordingly, we need to control distribution of malicious data to assure the integrity of content. Therefore, we consider the attack types of malicious nodes and evaluate the integrity of content. The type of attackers in the simulation are,

TABLE III
Simulation environment.

Network model	BA model
number of node	1000
maximum number of node	1100
minimum number of node	900
number of distinct files	100
ratio of malicious nodes	10%
number of divided decryption key n	20
threshold k	18

(1) Malicious node responds to every query with a fake data.
(2) Malicious node acts like a reliable node, but it tries to send a fake data when it gets enough reputation.
(3) Malicious node sends normally data to some nodes, but sends fake data to others.
(4) Malicious node sends normally data, but occasionally send fake data.
(5) Malicious node sends fake data and responds fake reputation when reputation of other malicious node requested.

Simulation environments are shown in Table III. Network model applies BA model which is an algorithm for generating random scale-free network. In this simulation, firstly, we construct network using BA model. Also, nodes have randomly incidence of entry and exit. At this time, nodes cannot entry such as exceed maximum node number and cannot exit such as fall below minimum node number. Also, owner cannot exit.

After network constructed using BA model, we distribute contents on network. Each content is distributed ten nodes that randomly selected on network. Then, in proposed method, decryption key is divided into n, and divided data are distributed n nodes that randomly selected on network.

After distribution of contents, normally nodes generate random request to contents. In this simulation, the owner cuts off generation of authentication, and requests node received authentication because only normally nodes generate the request. Also, in existing method[6], a request node received a decryption key. Then, the request node collects contents and a decryption key depending on each method. If malicious node received query, it sends data depending on the set-up attack type.

When request node finished transaction to collect contents and decryption key, downloads are success if it can decrypt the content. Also, downloads are failure if it cannot decrypt the content because of collected data includes fake data by a malicious node.

B. Result and consideration

Results of our simulation are shown in Figure 6-8. The x-axis of the figure shows the number of downloads. The y-axis of the figure shows ratio of failure to all downloads. Low ratio of failure to all downloads means each nodes download normally contents, therefore, it assures the integrity of content.

At attack type (1), (3) and (5), most downloads were failure from the beginning. This is due to receiving fake data from

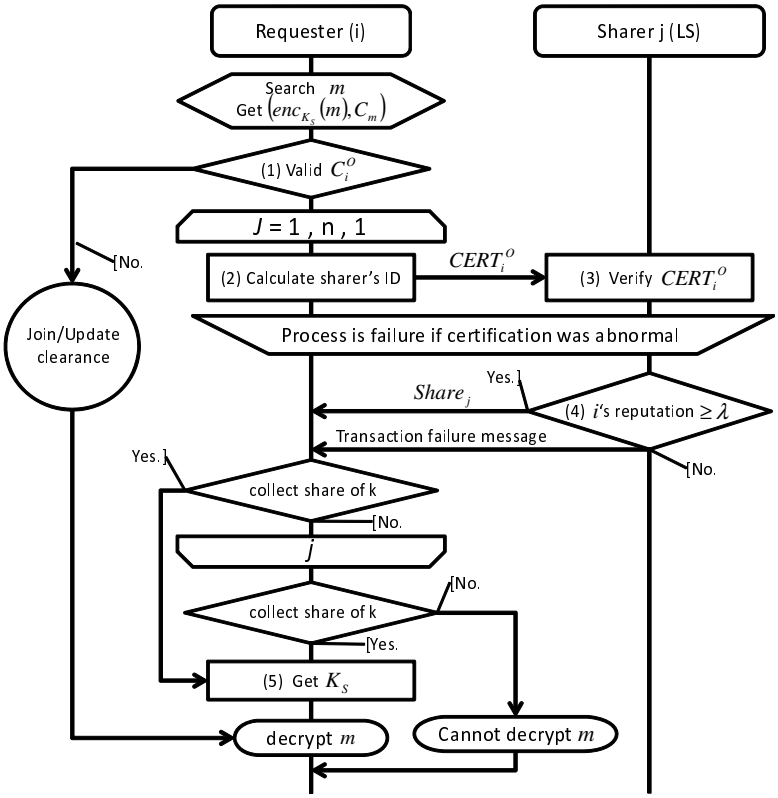

Fig. 5. Contents accessing process.

malicious nodes because they cannot collect reputation at beginning. However, in the proposed method, ratio of failure to all downloads is rapidly decrease if numbers of downloads are increased. On the other hand, in the existing method, ratio of failure to all downloads is almost constant rate. In the proposed method, nodes can control fake data in the early stage because the nodes collect reputation on all the reliable nodes.

Also, at attack type (2), most downloads were failure from the beginning in the proposed method. Then ratio of failure to all downloads is rapidly decrease if the number of downloads is increased. On the other hand, in the existing method, ratio of failure is slowly increased. In the proposed method, malicious nodes get reputation in the early stage because reputation value is frequently updated, therefore, request nodes receive fake data from malicious nodes. Node can control fake data as at attack type (1), (3) and (5) if the number of downloads is increased. In the existing method, malicious nodes don't send fake data in the early stage because it takes long to get reputation. However, ratio of failure is increased because normal nodes cannot collect accurate reputation of malicious nodes.

At attack type (4), ratio of failure is comparable range in both methods when malicious nodes start to send fake data. However, in the proposed method, ratio of failure is decreased in the early stage than the existing method. This is due to

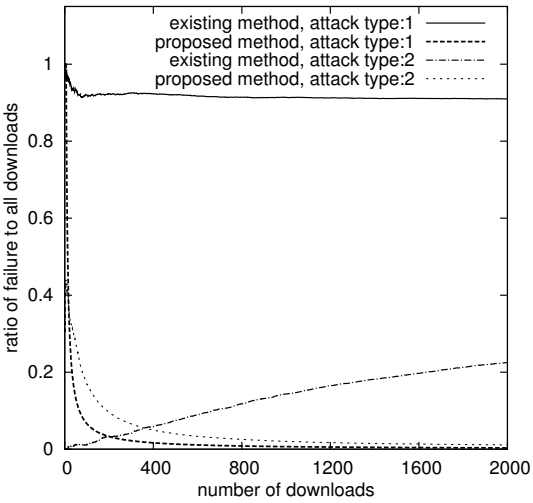

Fig. 6. Simulation results, attack type:(1),(2).

controlling fake data from malicious nodes because nodes collect reputation on all the reliable nodes as in the case of other attack types.

From these results, the proposed method assures the integrity of contents than the existing method.

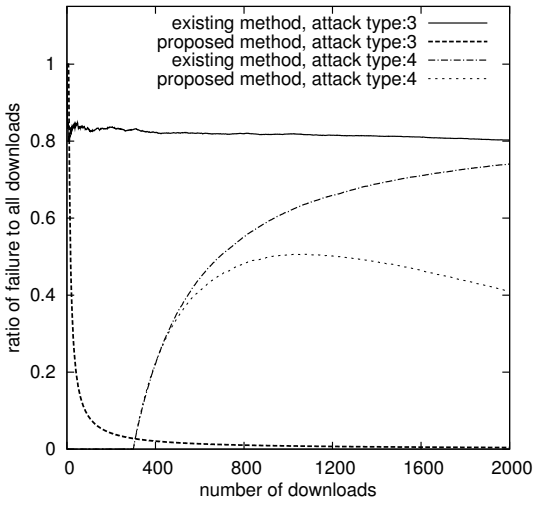

Fig. 7. Simulation results, attack type: (3),(4).

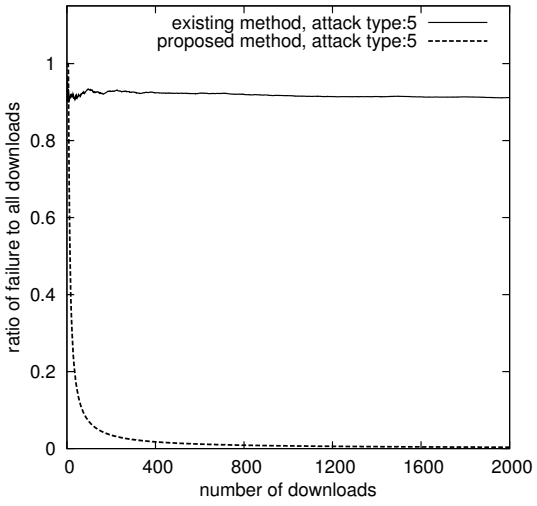

Fig. 8. Simulation results, attack type: (5).

V. Conclusion

In this paper, we proposed a secure content distribution system in a P2P network. An owner generates a content certificate which assures the integrity of content. The content certificate is generated based on receiving a multi-signature from a subset of nodes. A list of reliable nodes is generated based on reputation, which depends on past transactions. After generation of a content certificate, the owner encrypts the content by using a common key cryptosystem and distributes a decryption key by using a secret sharing scheme. A node requests generation of an access certificate from the owner. If the owner assents to the request based on reputation, it generates an access certificate which includes a signature and decryption key information. The request node obtains the decryption key by using the access certificate and then can access the content.

We evaluated the integrity of content. We found that the proposed method assures the integrity of contents than the existing method.

References

[1] Esther Palomar, Juan M.E. Tapiador, Julio C. Hernandez-Castro, Arturo Ribagorda,"Secure content access and replication in P2P networks", Computer Communications 31,pp.266-279(2008).

[2] Blakley, G. R.,"Safeguarding cryptographic keys", Proceedings of the National Computer Conference 48, pp.313-317(1979).

[3] A.Samir, "How to Share a Secret", communication of the ACM, Vol.22, No.11, pp.612-613(1979).

[4] Stakhanova N., Basu S., Wong J., Stakhanov O., "Trust Framework for P2P Networks using Peer-Profile based Anomaly Technique", Proseedings of the 2nd International Workshop on Security in Distributed Computing Systems, pp.203-209(2004).

[5] Chopra K, Wallace W, "Trust in Electronic Enviroments",Proceedings of 36th Annual Hawaii International Conference on System Sciences,pp.331-340(2003).

[6] A. A. Selcuk, Ersin Uzun, M. R. Parriente, "A Reputation-Based Trust Management System for P2P Networks", IEEE/ACM International Symposium on Cluster Computing and the Grid CCGrid, pp.251-258(2004).

[7] Withby A., Josang A., Indulska J, "Filtering Out Unfair Ratings in Bayesian reputation Systems", Proseedings of the 3rd International Joint Conference on Autonomous Agents and Multi Agent Systems, pp.48-64(2004).

[8] Guha R., Kumar R., Raghavan P., Tomkins A., "Propagation of Trust and Distrust",Proceedings of the 13th International Conference on World Wide Web, pp.403-412(2004).

[9] Kamvar S., Schlosser M., Garcia-Molina H., "The EigenTrust Algorithm for Reputation Manegement in P2P Networks", Proceeding of the 12th International Conference on World Wide Web,pp.640-651(2003).

[10] Jianguo Chen., Huijuan Lu., Bruda S.D., "A Reputation-Based Approach for Countering Vulnerabilities in P2P Networks", e-Business and Information System Security (EBISS), 2010 2nd International Conference, pp.1-4(2010).

[11] Jianli Hu., Xiaohua Li., Bin Zhou., Yonghua Li., "A Reputation Based Attack Resistant Distributed Trust Management Model in P2P Networks", Electronic Commerce and Security (ISECS), 2010 Third International Symposium, pp.237-241(2010).

[12] Satsiou A., Tassiulas L., "Reputation-Based Resource Allocation in P2P Systems of Rational Users", Parallel and Distributed Systems, IEEE Transactions, pp.466-479(2010).

[13] Xiaoning Jiang, Lingxiao Ye, "Reputation-Based Trust Model and Anti-attack Mechanism in P2P Networks", Networks Security Wireless Communications and Trusted Computing (NSWCTC), 2010 Second International Conference, pp.498-501(2010).

[14] Lei Yang, Zhiguang Qin, Hao Yang, Can Wang, ChaoSheng Feng, "A reputation-based method for controlling free-ride in P2P networks", Communications, Circuits and Systems (ICCCAS), 2010 International Conference, pp.249-253(2010).

[15] Tauhiduzzaman M., Mea Wang, "A system analysis of reputation-base defences against pollution attacks in P2P streaming", Performance Computing and Communications Conference (IPCCC), 2012 IEEE 31st International, pp.152-161(2012).