*International Journal of Engineering & Technology*, 7 (4.39) (2018) 539-544

International Journal of Engineering & Technology

*Website: www.sciencepubco.com/index.php/IJET*

***Research paper***

# Efficient Big Data-based Access Control Mechanism for IoT Cloud Environments

**Yoon-Su Jeong[1], Yong-Tae Kim[2*], Gil-Cheol Park[3]**

*[1]Dept. of Information and Communication Convergence Engineering, Mokwon University, 88, Doanbuk-ro, Seo-gu, Daejeon, 35349,Republic of Korea*

*[2,3]Dept. of Multimedia, Hannam University, 70 Hannam-ro, Daeduk-gu, Daejeon, 34430, Republic of Korea*

**Corresponding author E-mail: ky7762@hnu.kr*

## Abstract

**Background/Objectives**: Recently, the data using in the internet had processed through the network every day, and cloud services related to IoT are increasing rapidly. In particular, the cloud service related to IoT has been transformed into an era in which data is generated and processed by an individual centered around the enterprise in the past. However, the use of mobile phones and IoT technology had diversified, and the demand for computational cost and accuracy has increased.

**Methods/Statistical analysis**: we propose access control method based on big data that can process various attributes of data in block in IoT cloud environment. The proposed scheme aims to minimize the service latency of users by extracting the security parameters δ of each data by attribute and converting them into pairs with polynomials. In the performance evaluation, we had fined that the data processing time was 7.2% higher than previous scheme and the data processing rate was 9.7% higher than previous scheme. The accuracy according to the type and size of the different data improved by 18.1%. IoT cloud server and user communication delay was 8.5% higher than previous scheme. Finally, the server overhead reduced by 5.8%.

**Findings**: We propose a method that can access verified data without delaying the data by constructing data into subnets and then applying the security parameter δ of the data constituting each subnet to n-bit and applying it to the polynomial coefficients.

**Improvements/Applications**: In future research, proposed scheme can be applied to various services related to large-scale data access in the cloud environment.

***Keywords:*** *Access Control, Algorithm, IoT, Big Data, Cloud Service.*

## 1. Introduction

In the cloud environment where IoT devices are used recently, various technologies and services such as statistical data using bioinformatics, medical technology, technology management, marketing, weather information, security management, and Google translation are applied to various fields, It is in the limelight [1]. In particular, Big Data in the IoT cloud environment can generate, collect and analyze various kinds of data, and it is a service that can develop the diversified modern society more efficiently. Therefore, this is possible that the customized information provided, managed and analyzed for each personalized modern society member.

As the IOT is combined with Big Data, diversified services are provided, and the requirements for data accuracy and processing technology due to big data increase are increasing [2,3]. Big Data used in various fields of our society including medical, economic, and scientific fields. However, different methods of generating, collecting, and analyzing different types of big data are different from each other, and the processing speed for providing customized information has become one of the most important requirements [4].

In recent researches related to IoT cloud environment, IoT related standardization work is proceeding simultaneously in various international standards organizations such as IPSO Alliance, OMA, oneM2M, Zigbee, and IETF. The Internet Protocol for Smart Object (IPSO) Alliance has been conducting activities related to education, publicity and research related to the use of Internet (IP) technology for connecting Smart Objects[5]. 6LoWPAN, ZigBeeIP and CoAP (Constrained Application Protocol) are studied as a technology for transmitting sensing data information measured by sensors to the gateway. In the backend part, IoT Device is a standard for efficiently managing sensed data and resources IETF, IPSO, and oneM2M [6, 7]. In addition, a message standard for transferring measured data from IoT devices to the platform is under development in OMA [8].

In this paper, we propose an efficient big data scheme based access control that reflects the characteristics of data. The proposed scheme constructs data into a subnet so that the data processed in the IoT cloud environment can be stored in various servers. The configured subnets can verified without delaying the data by applying the security parameter δ of the data constituting each subnet to n-bit and applying it to the polynomial coefficients. In particular, the proposed method improves efficiency by using the vector generated by polynomial coefficients to control the access of big data to improve the verification speed of data used in IoT cloud environment. In addition, the proposed method uses the polynomial coefficients to represent the big data as a vector, and converts the vector to minimize illegal data access.

The proposed method has three purposes as follows. First, the data stored in the server organized into a hierarchical subnet to the data access desired by the user. Second, probability-based clustering

property information is extracted to improve data accuracy among big data of different kinds (size, usage, type, etc.). Third, the service delay time minimized by using the polynomial of the vector processed by polynomial coefficients.
The composition of this paper is as follows. In Chapter 2, we will look at big data concept and existing research. In Section 3, we propose a secure and verifiable access control scheme to improve data processing. In Section 4, we compare and evaluate the proposed scheme and the previous scheme. Finally, in Section 5, we conclude the paper.

# 2. Related Works

## 2.1. Big Data

The big data used in the cloud service is not only numbers but also images [1]. Recently, data can be easily stored in cyberspace regardless of time and place. This phenomenon is due to the explosion of digital information due to the proliferation of M2M (Machine to Machine).
The textual information circulated in the blog or the SNS is not only the tendency of the person who writes the text but also the connection relation of the communication partner Can be analyzed. Data is being used not only in science but also in all social sectors such as bioinformatics, medical technology, technology management, marketing, weather information, security management, and Google translation.
Big data have characteristics such as 3V(volume of data, velocity of data generation, and varieties of shape). Although vast amounts of data are used as an important resource that dominates the competitiveness of the nation, the paradigm shifts in terms of quality and diversity as well as the amount of data compared to the past.
Big Data has made it possible to analyze large-scale customer information in a short period of time by using technology such as distributed processing method. In Big Data, software and hardware utilize open source Hadoop, analysis package R and distributed parallel processing technology, cloud computing, etc., so do not build expensive data warehouses based on existing expensive storage and databases. It is also possible to operate the system effectively without [4].
Hadoop distributed file system refers to a file system that utilizes unreliable hardware to support very large data streaming rather than accessing it [9]. The Hadoop distributed file system can be stored on any disk without file size limitation, and block abstraction simplifies the storage subsystem and improves efficiency.
The Hadoop distributed file system is intended for high data throughput and aims to reduce costs by actively utilizing existing systems without using hardware that is highly reliable. Since the metadata of the file system managed in the memory of the node, the number of files depends on the memory size. The HDFS block configured to be larger than a conventional disk block to minimize the search cost. If the block is large enough, it can allocate more time to the data transfer time than the time to search the start point of the block from the disk [10].

## 2.2. Big Data Detection Algorithm

The Big Data - related detection algorithms that have been studied so far focus on analyzing small - sized data such as microblogs [11]. However, the CELF algorithm among the big data detection algorithms is one of the most widely known algorithms for detecting large data [12]. The feature of the CELF algorithm is to select subnets so that all events can detected as completely as possible. However, the CELF algorithm has not yet been able to optimize the mixed-interger in selecting the optimal subnet.
K. Chen et. al technique has proposed a method to detect key events online in resource constraints [13]. However, this method has a problem in that a subnet size selected to be small to efficiently detect small size data such as a micro blog. D. Kempe et. al technique selects subnets of nodes that have a large influence on data search and represents them as probability values [14]. Even if this technique does not generate events that affect the search, it greatly affects the probability of creating a subnet.

## 2.3. Previous Research

In order to process various types of data processed on the Internet, S. O. Park. et. al proposed a method to support a plug-in web browser [15]. This method characterized by having compatibility between various web browsers through a plug-in. This technique also supports inter-server communication by registering a local server that can invoke a local plug-in using the compatibility framework (F4C). However, this technique has a problem in that it can request and respond to a remote server without permission of the user.
To provide cloud services seamlessly, T. Joelsson. et. al proposed a method of interworking between smartphone and mobile web browser [16]. This technique characterized by making the relevant components of smartphones and mobile web browsers available to the local web server. However, this technique is disadvantageous in that a local web server can't execute a proxy-like operation after downloading scripts operated on a web server remotely.
For evaluating the big data stored in the cloud server. B. B. Miao et. al method uses an unordered evaluation method of Big Data [17]. In this way, we analyze the time series of big data and find tradeoff between big data calculation speed. However, this method is problematic in that it only finds the calculation speed for arbitrary big data stored in the cloud server.
C. Barbieru et. al method allows the Hadoop scheduler to run on servers running in a smart-city environment [18]. However, this method has a problem that the performance of the server is deteriorated when the data processing amount that can be processed by the server is exceeded.

# 3. Big Data Access Techniques to Improve Data Processing Speed

This section proposes a big data-based access control scheme that can minimize the processing time of many data processed in the cloud environment.

## 3.1. Overview

With the recent Internet development and popularization of mobile phones, hundreds of thousands of data are processed every day through the network. In the past, data generation and processing were done by companies, but recently, data is being generated and processed by individuals. Particularly, the purpose of use of data is diversified, and the size and storage location of data are increasing. In particular, there is a growing demand for computational cost and accuracy for different data types and sizes.
In this paper, we propose an efficient big data scheme based access control that reflects the characteristics of data processed in IoT cloud environment. The proposed method minimizes the processing time in the cloud server by probabilistically extracting the security parameter $\delta$ of each data after decentralizing various attributes of the data to be processed in block units. The proposed scheme constructs data into a subnet so that the data processed in the IoT cloud environment can be stored in various servers. The configured subnets can verified without delaying the data by applying the security parameter $\delta$ of the data constituting each subnet to n-bit and applying it to the polynomial coefficients. In particular, the proposed method uses polynomials and pairs of polynomial coefficients to improve the verification speed of data used in IoT cloud environment.

#### 3.1.1 System Model

The proposed scheme considers a cloud storage system with hierarchical multiple attributes. The system model used in the proposed scheme consists of components as following.

• Authorities

If all the privileges used in the proposed scheme are independent of each other, They manage the attributes of the user requesting the service. It also creates a secret / public key pair for each attribute used in the subnet. The secret key is classified into subnets to the characteristics (feature, attribute, etc.) of the big data according to the attribute information of the user, and the secret key is generated by assigning a polynomial coefficient to the classified data.

• Cloud server

The cloud server performs access control of the user requesting the service based on the access control policy.

• Data Owners

The data owner is responsible for defining access policies and data encryption / decryption to suit the cloud server policy before hosting the data in the cloud. The data owner also checks to see if the cloud server policy has been updated.

• User

Users who want to access the cloud service are assigned access identifiers and can access the server freely.

#### 3.1.2. Framework

The access control framework of the proposed scheme is as follows.

· GlobalSetup (δ)→ GP

GlobalSetup (δ) is inputs the security parameter δ.

· AuthSetup (GP, AID)→ (SK, PK)

The permission setting receives the global parameter GP and the rights recognizer and outputs a secret / public key pair.

· KeyGen (GID, GP, A, $SK_{AID}$) → $SK_{GP,AID}$

The key generation uses the secret key $SK_{AID}$ corresponding to the user global identifier to generate the global secret key $SK_{GP,AID}$. At this time, the global parameter GP and the authority information A are used for the hierarchical access right of the big data processed in the subnet.

· Encrypt (PK, GP, bd, AP) → CT

According to the cloud server access policy AP, the cipher text CT is generated by encrypting the big data bd with the public key PK together with the global parameter GP.

· Decrypt (CT, GP, $SK_{GP,AID}$) → bd

The decryption process obtains the big data bd using the global secret key $SK_{GP,AID}$ together with the global parameter GP to decrypt the ciphertext CT. At this time, the access policy of the cloud server is checked and if access policy is inconsistent, access is prevented.

· UKeyGen (PK, A, $A'$, bd) → UKey

The key renewal process encrypts the existing big data access right A and the updated access right $A'$ of the cloud server with the public key PK together with the big data to generate the key UKey necessary for updating the big data access right in the cloud server,

· UAuthority (CT,UKey ) → $CT'$

The authority renewal process uses the ciphertext CT and the renewal key UKey to generate the updated ciphertext $CT'$.

### 3.2. Notations

Table 1 summarizes the notations used in the proposed scheme.

**Table 1:** Notation

| Parameter | *Notation* |
|---|---|
| δ | Security Parameter |
| GP | Global Parameter |
| AID | Authority Identifier |
| A | Authority |
| $A'$ | Updated authority |
| SK | Security Key |
| PK | Public Key |
| AP | Access Policy |
| CT | Cypher Text |
| bd | Big data |
| UKey | Updated Key |
| AM | Authentication message |
| IV | Intermediate value |

### 3.3. Generate Subnet Polynomial Coefficients

The proposed method constructs subnets as shown in Equation (1) to generate the polynomial coefficients.

$$\mathrm{AP(x)} = \sum_{i=0}^{n-1} AP_i x^i \quad (1)$$

Where, i denotes the number of data constituting the subnet, and $AP_i$ denotes a vector component of the polynomial.

At this time, the polynomial constituting the subnet calculated as AP in the equation (2).

$$\mathrm{AP} = (AP_0, AP_1, \ldots, AP_{n-1}) \quad \mathrm{n} = 0, 1, \ldots, \mathrm{n\text{-}1} \quad (2)$$

Where, $AP_{n-1}$ means an element constituting a vector.

The proposed scheme constructs a pair of polynomial coefficients and probability values for subnets. In this case, the similarity of the extracted data through the probability value is effective for data search.

The proposed scheme constructs subnets by combining high probability data. At this time, data configured to be stochastically layered according to the subnet configuration. The data are all highly related data, and the subnet is composed mainly of the data having a high association value.

### 3.4. Polynomial-based Property Access Control to Improve Big Data Processing

Subnets classified according to characteristics (characteristics, attributes, types, functions, etc.) of data stored in the cloud server to improve the processing speed of large data processed by the cloud server. The attribute access control is process for improving the processing speed is composed of four processes including a system initialization process, a key generation process, a data authentication process, and a policy management and update process.

#### 3.4.1 System Initialization Process

This section shows the initialization process of the system for big data processing. For the system initialization process, we first assume that there are N users who receive big data service. The user $U_i$ that can accurately verify data among N user U is the same as $\widehat{U} \subseteq$ U as in Eqs. (3) to (4).

$$\mathrm{U} = \{u_1, u_2, \ldots, u_N\}, \ \mathrm{i} \in [1, \mathrm{N}] \quad (3)$$

$$U_i = \{\widehat{U} \subseteq \mathrm{U} \mid \mathrm{i} \in [1, \mathrm{N}]\} \quad (4)$$

Where, the user $U_i$(i∈[1, N]) sets the data set $D_i$ by sampling the data d in N pieces.

The data set $D_i$ is obtained by sampling the data d by N as shown in equation (5), and then the data set is obtained as $D_i$(i∈[1, N]) as in equation (6).

$$\mathrm{L} = \{d_1, d_2, \ldots, d_N\}, \ \mathrm{i} \in [1, \mathrm{N}] \quad (5)$$

$D_i = \{D_i \subseteq L \mid i \in [1, N]\}$ (6)

Where, $d_i$ denotes verifiable data that can be securely provided.
For big data access control, the data set constructed like $D_i \subseteq L$ as in Eq. (7), and the vector component value of the polynomial expressed as $\bar{d}$of Eq. (8).

$D_i = (p_1, p_2, ..., p_n)$ , $i \in [1, N]$ (7)

$\bar{d} = \{ dp_i \in Z \mid d_j \sim dp_i \}$, $1 \le i \le N$, $1 \le j \le N$ (8)

Where $p_i$ ( $i \in Z$ ) denotes the vector component value of the polynomial. $\bar{d}$ is a set of vector component of all polynomials related to$p_i$. $dp_i$($i \in [1, N]$) means the data probability included in the dataset.
$\bar{d}$ is the vector component value of the polynomial, as shown in equation (9), which represents the binary probability information.

$$P_i = \begin{cases} 1 & if\ d_i participated \in dataset \\ 0 & otherwise \end{cases} \quad (9)$$

Where, $P_i$ is represented by a data probability of 0 and 1.
The IoT cloud server assigns the information constituting the subnet to $d_i$. The set of vector components of the polynomial information $\vec{D}$ has the highest probability of the information and characteristics of the corresponding vector information as equation (10).

$CI_i = H(\vec{D}, \bar{d})$, $1 \le i \le n$ (10)

The linkage information $CI_i$ generated by Eq. (10) constructed as Eq. (11) with the data. At this time, the user $U_i$ monitors the information of Equation (12) in real time and uses it to search for big data.

$\sum_{i=1}^{n} H(Data, CI_i)$ (11)

**3.4.2. Key Generation Process**

In this section, the process of creating a key to protect the data generated by IoT devices in a cloud environment when a user attempts to transfer data to the cloud server is shown in the following three steps.
· Step 1: In order to generate a key for authenticating information for transmitting and receiving data among IoT devices operating in a cloud environment, the proposed method constructs data into a subnet and then generates a security parameter δ of the data constituting each subnet do. The generated security parameter δ converted into an n -bit block so that it can applied to the polynomial coefficients.
· Step 2: The cloud server receives the security parameter δ of the IoT device and applies it to the key generation function as Equation (12).
$Keygen[\delta] \rightarrow \lambda$ (12)
Where, λ is the linkage information between IoT devices generated through the key generation process.
· Step 3: The cloud server checks the IoT device that wants to upload data to the cloud server and checks whether the security parameter δ of the data has been converted into the n-bit block normally. If the check result is a normal result, the cloud server executes the symmetric encryption algorithm by applying the linkage information λ between the IoT devices to each n-bit block. The cloud server transmits the IoT device using the number of equipment installed. The cloud server divides the security parameter of the data δ into n-bit blocks again when the abnormal result obtained.

**3.4.3 Data Authentication Process**

In the proposed scheme, polynomial authentication performed to perform secure and verifiable access control of big data. The reason for performing polynomial authentication is to improve the processing speed of big data without load of IoT cloud server.
In order to perform authentication, we define polynomial as $A_{MN}$ in Eq. (13). At this time, $A_{N_1}$ is a value for the first digital signature, so that a value of 1 always appears. Also, $A_{MN}$ is the last value of the data, and it is set to 1 for the digital signature.

$A_{(n-1)(k-1)} + A_{(n-1)k}$ (13)

In Eq. (13), if a positive integer with n and k equal to $n \ge k$ is established, then a polynomial like Eq. (14) is established.

$$\binom{n+1}{k} = \binom{n}{k-1} + \binom{n}{k} \quad (14)$$

In Eq. (15), $a_i$ corresponds to the $(n+1)^{th}$ row of data.

$$(x+y)^n = a_0 x^n + a_1 x^{n-1} y^1 + a_2 x^{n-2} y^2 + \cdots + a_n y^n \quad (15)$$

In the proposed scheme, the access control of the big data facilitated by allowing the data to be continuously used by including the authentication information.

**3.4.4 Policy Management and Renewal Process**

This process shows the process of dynamically processing the access control required for updating the data by checking the access policy of cloud server.
The encrypted data to be updated by the user is stored in advance in the cloud server. If necessary, the user decrypts the cipher text in order to download and obtain the plaintext, re-encrypts the plaintext according to the new access policy, and uploads it to the cloud server.
The new data access policy defined as Eq. (16).

$AP'(x) = AP + \sum_{i=0}^{n-1} AP_i' x^i{}_i$ (16)

Where, $\sum_{i=0}^{n-1} AP_i' x_i$ means a new access polynomial selected by the user who owns the data, and n updates the access policy of the cloud server.
The user generates an intermediate value IV using the access policy AP and $AP'$ and the authentication messages AM and $AM'$as in equation (17).

$IV = H(AP * AM) \,||\, H(AP' * AM')$ (17)

Where, AP and AM are the access policies and authentication messages before updating the data of the cloud server, and $AP'$ and $AM'$are the access policies used for updating the information stored in the server. Authentication message.
The server generates the intermediate value $IV'$as shown in Eq. (18) to verify that the user's ciphertext has been updated normally by the new ciphertext policy and delivers it to the user.

$IV' = \prod_{i=1}^{n} IV_i^{n-1}$ (18)

When Expression (18) transmitted to the user, the user performs the process of generating the hash information of the big data for accessing the update data stored in the cloud server as Expression (19).

$H(bd') = H(IV' \,||\, H(AP' * AM'))$ (19)

# 4. Evaluation

## 4.1. Data Processing Time

Figure 1 shows the data processing time of the server. In Fig. 1, the proposed method compares the search processing speed of a large amount of big data processed by the server according to the characteristics (types, functions, features, attributes, etc.). The average retrieval processing time was 7.2% higher than the conventional method. The proposed scheme does not use the additional encryption/decryption process by generating the polynomial coefficients by subnetting according to the characteristics of the data when uploading the data to the cloud server. In addition, the proposed method processes the data according to the access policy of cloud server. Through this process, the proposed method does not go through additional processing of data, so even if uploading a large amount of data, the search processing time does not increase more than the existing method.

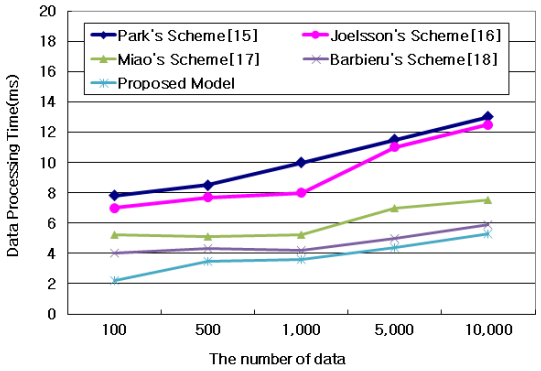


**Figure 1:** Data processing time

## 4.2. Overhead

Figure 2 shows the overhead of the cloud server when the server processes data generated by IoT devices. As shown in Fig. 2, the proposed method is 5.8% lower than the conventional method.

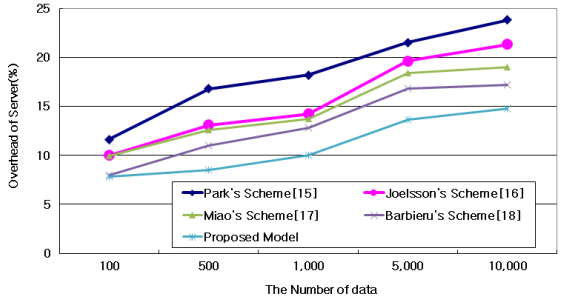


**Figure 2:** Overhead of Server

In the result shown in Fig. 2, the proposed scheme reduces the overhead of the cloud server because it converts the vectors into polynomials and pairs.

## 4.3. Data Process Rate

Figure 3 compares the data process rate of the cloud server with the previous scheme. In Figure 3, the data stored in the server is composed of hierarchical subnets according to the characteristics of the data. Therefore, the data processing rate of proposed scheme per unit time is 9.3% higher. The proposed scheme can securely retrieve the desired data to verify the data securely and extracts global attribute information. In addition, the data processing rate of the cloud server is higher than previous scheme because the vector represented by the polynomial coefficient converted into a pair with the polynomial.

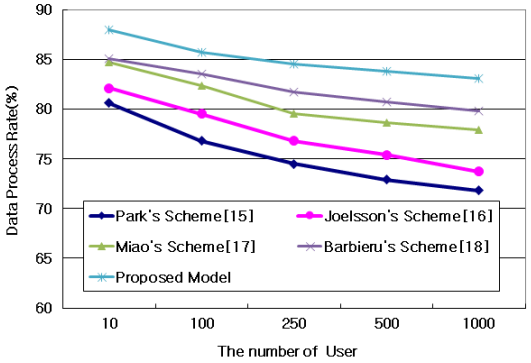


**Figure 3:** Data Process Rate of Server per Time

## 4.4. Data Accuracy

Figure 4 shows the data accuracy according to the type and size of the different data stored in the cloud server. In Fig. 4, the data accuracy of proposed scheme is 18.1% higher than the previous scheme. This result is the result of the proposed method extracting the security parameter δ of each data and linking it to the probability value. Since the security parameter δ of the data constituting each subnet converted into n -bit and applied to the polynomial coefficient, the accuracy improved because the data verified safely and accurately without delay time of data. In particular, the proposed method reflects the polynomial and paired vector generated by polynomial coefficients to the control policy of cloud access to improve the verification speed of data used in IoT cloud environment. Therefore, the accuracy of the data improved compared to the existing technique.

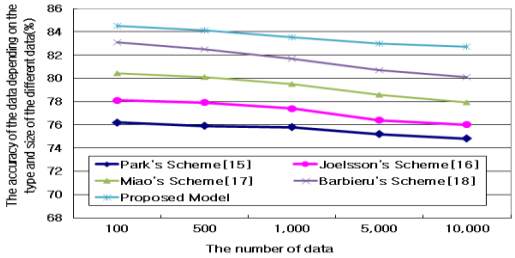


**Figure 4:** Data accuracy

## 4.5. Communication Latency

Figure 5 shows the communication delay time between user cloud servers through policy management and update of cloud server so that users can securely verify data in IoT cloud environment. As shown in Fig. 5, according to the cloud policy, by extracting the vector using the polynomial coefficient of the data identification information, the delay time for the user to access the cloud server reduced 8.5% lower than the previous scheme. The data stored in the cloud server is composed of subnets. The security parameter δ of the data is applied to the polynomial coefficients.

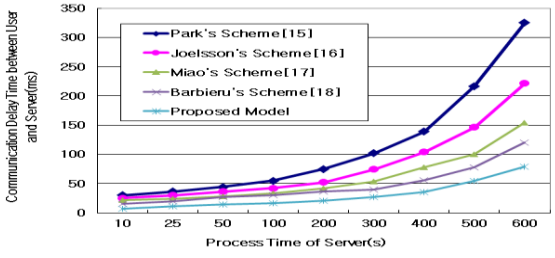


**Figure 5:** Communication latency

## 5. Conclusion

With the development of the Internet, the demand for IoT cloud services has increased recently. However, the cloud service does not provide enough data processing technologies compared to the development of the application used in the mobile phone. In this paper, we propose an efficient big data scheme based access control that reflects the characteristics of data processed in IoT cloud environment. The proposed scheme extracts the security parameter δ of each data stored in the server and minimizes the process time of data composed of subnet. The proposed scheme improves the efficiency of cloud access control policy management and update to improve the verification speed of data used in IoT cloud environment. . In the performance evaluation, we had fined that the data processing time was 7.2% higher than previous scheme and the data processing rate was 9.7% higher than previous scheme. The accuracy according to the type and size of the different data improved by 18.1%. IoT cloud server and user communication delay was 8.5% higher than previous scheme. Finally, the server overhead reduced by 5.8%. In future research, we plan to apply the proposed method to the actual environment.

## Acknowledgment

This paper has been supported by 2018 Hannam University Research Fund.

## References

[1] Kan Y, Xiaohua J. Kui R, Secure and Verifiable Policy Update Outsourcing for Big Data Access Control in the Cloud, IEEE Transactions on Parallel and Distributed Systems, 2015, 26(12), pp. 3461-3470.
[2] Chen H S, Bhargava B, Fu Z C, Multiables-Based Scalable Access Control for Big Data Applications, IEEE Cloud Computing, 2014, 1(3), pp. 65-71.
[3] Hu C Q, Li W, Cheng X. Yu J. Wang S, Bie R, A Secure and Verifiable Access Control Scheme for Big Data Storage in Clouds, IEEE Transactions on Big Data, 2018, 4(3), pp. 341-355.
[4] Zeng W, Yang Y, Luo B, Access control for big data using data content, Proceedings of the 2013 IEEE International Conference on Big Data, 2013, pp.45-47.
[5] Aris I B, Sahbusdin R K Z, Amin A F M, Impacts of IoT and big data to automotive industry, Proceedings of the 2015 10th Asian Control Conference, 2015, pp. 1-5.
[6] Barreto, F M, Duarte, P A. de S. D, Maia M E F, Andrade R M. de C, Viana W, CoAP-CTX: A Context-Aware CoAP Extension for Smart Objects Discovery in Internet of Things, Proceedings of the 2017 IEEE 41st Annual Computer Software and Applications Conference, 2017, 1, pp. 575-584.
[7] Wu C W, Lin F J, Wang C H, Chang N, OneM2M-based IoT protocol integration, 2017 IEEE Conference on Standards for Communications and Networking , 2017, pp. 252-257.
[8] Amur H, Cipar J, Gupta V, Ganger G R, Kozuch M A, Schwan K, Robust and flexible power-proportional storage, Proceedings of the 1st ACM symposium on Cloud computing, 2010, pp. 217-228.
[9] Merla P, Liang Y, Data analysis using Hadoop MapReduce environment, Proceedings of the 2017 IEEE International Conference on Big Data, 2017, pp. 4783-4785.
[10] Strang K D, Sun Z, Meta-analysis of big data security and privacy: Scholarly literature gaps, Proceedings of the 2016 IEEE International Conference on Big Data, 2016, pp. 4035-4037.
[11] Revathy P, Mukesh R, Analysis of big data security practices, Proceedings of the 2017 3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), 2017, pp. 264-267.
[12] Shen P, Zhou Y, Chen K, A Probability based Subnet Selection Method for Hot Event Detection in Sina Weibo Microblogging, IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 2013, pp. 1410-1413.
[13] Chen K, Zhou Y, Zha H, He J, Shen P, Yang X, Cost-Effective Node Monitoring for Online Hot Event Detection in Sina Weibo, Proceedings of the 22nd international conference on World Wide Web, 2013, pp. 107-108.
[14] Kempe D, Klenberg J, Tardos E, Maximizing the spread of influence through a social netowrk, Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, 2003, pp. 137-146.
[15] Park S O, The Framework for Providing Compatibility to various Web Browser Plug-ins, Master's Thesis, KAIST, 2009.
[16] Joelsson T, Mobile Web Browser Extensions, Master of Science Thesis, KTH Information and Communication Technology, 2008.
[17] Miao B B, Jin X B, Compression processing estimation method for time series big data,Processing of the 27th Chinese Control and Decision Conference(2015 CCDC), 2015, pp. 1807-1811.
[18] Barbieru C, Pop F, Soft Real-Time Hadoop Scheduler for Big Data Processing in Smart Cities, Proceedings of the 2016 IEEE 30th International Conference on Advanced Information Networking and Application (AINA), 2016, pp. 863-870.