International Journal of Engineering & Technology, 7 (2.12) (2018) 18-24

International Journal of Engineering & Technology

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

Research Paper



Efficient key distribution protocol for mobile devices in cloud environments

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

[1] Dept. of Information Communication Engineering, Mokwon University, 88, Doanbuk-ro, Seo-gu, Daejeon, 35349,Republic of Korea
[2] Dept. of Multimedia, Hannam University, 70 Hannam-ro, Daeduk-gu, Daejeon, 34430, Republic of Korea
**Corresponding author E-mail: bukmunro@mokwon.ac.kr*

Abstract

Background/Objectives: Recently, as the fourth industrial revolution has emerged, cloud computing services have been attracting attention for efficient use of Internet-based computing resources. Among the cloud computing services, even if the data processed by the mobile device is encrypted after being stored in the server, the confidential information can be leaked. Therefore, there is a need for the key generation for data encryption and decryption.
Methods/Statistical analysis: In this paper, we propose a key distribution protocol that enables mobile devices to securely encrypt and decrypt keys in an efficient manner in a cloud environment. The main purpose of the proposed protocol is to maximize the efficiency and cost reduction of key generation, which can securely transmit and receive data, in a situation where the size of data used in the cloud environment and the storage location are increasing. As a result of the performance evaluation, the proposed method improved the authentication processing time by 4.1% on average compared with the existing protocol, and the average throughput rate of the server per unit time was 6.5%. In addition, the communication delay time between the authentication server and the mobile device improved by 9.3% on average, and the authentication overhead of the server was 11.5% lower than that of the conventional method.
Findings: In order to solve this problem, the proposed protocol can solve the security problem of the mobile device because it can receive the authentication through the one - way hash function and the XOR operation using the encrypted data using the session key.
Improvements/Applications: In future studies, we will apply the proposed protocol to the actual environment based on the results of this study and compare it with the results obtained from the theoretical studies.

Keywords: *Cloud Computing; Authentication Protocol; Distribution Process; XOR; Mobile Device; Key Generation & Distribution.*

1. Introduction

Recently, as the social requirements for the 4th industry have increased, there has been an increasing interest in cloud computing services capable of servicing computing resources such as hardware and software through the Internet [1]. Particularly, it has attracted a great deal of attention from the user that the computing resources existing in different physical locations can be integratedly managed through the virtualization technology.
Cloud services have been provided to companies in the past, but in recent years, services have been offered to individuals with the integration of Internet (IoT, Internet of Things) technology. However, as the purpose of using the cloud service is diversified, there is an increasing demand for the security of the data stored in the cloud environment [2].
Data stored in cloud computing is stored in the cloud server after encryption. However, even if confidential data is encrypted and stored, the encrypted data is stored in the cloud server, so confidential information can be leaked during the virtual device and cloud computing services [3].
Most cloud computing environments that have been in operation so far have the advantage of being able to browse and modify their own documents anytime and anywhere. However, personal information may be leaked if the server is hacked. In a cloud computing environment, a 2-factor authentication method that ensures high security is required in order to safely store data on a server.
Security attacks on virtual and cloud platforms that are used in cloud computing environments are very easy, but protection is very difficult. As enterprises adopt security technologies for them, IT managers who have to protect the important data of the enterprise are burdened even more. Patching a massive virtualization server is not an easy task, and it can provide hackers the ability to steal servers, interfere with traffic, and steal data from vulnerable systems [4].
In this paper, we propose an efficient key distribution protocol that can securely distribute keys that can be used to securely transmit and receive data in a mobile environment. The most important difference between the proposed protocol and the existing protocol is that it does not use additional cryptographic algorithms when performing key distribution. In addition, the proposed protocol uses the previously generated session key SK to prevent a mobile device at a different physical location from illegally exploiting confidential information. Since the session key SK does not provide any information form other than the authentication server to the third party, it can safely provide confidential information of the mobile device by encrypting / decrypting important information between the server and the mobile devices. In addition, since the proposed protocol uses a one-way hash function and XOR operation to receive a service existing in a server when a mobile device accesses a specific server from a remote place, it is excellent in terms of efficiency as well as cost reduction. In order to solve the security problem of mobile devices in the cloud computing, the proposed protocol distributes the authentication to the mobile devices existing in the cloud

Copyright © 2018 Yoon-Su Jeong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

environment and uses the integrated authentication system provided by the cloud platform for the external mobile devices.
The composition of this paper is as follows. Chapter 2 discusses cloud computing and cloud computing security research. In Section 3, we propose an efficient key distribution protocol for mobile devices operating in a cloud environment. In Section 4, performance and security evaluation of the proposed protocol are analyzed. Finally, Section 5 concludes the paper.

2. Related works

2.1. Cloud computing

Cloud computing is a service that enables users to access computing resources existing on the Internet anytime and anywhere by various methods according to their needs [1], [5], [6].
Components that make up the cloud computing environment include software, storage, and networking, as shown in Figure 1. Cloud computing can get as many resources as needed, regardless of location and time, if you want to. In order to provide smooth services to users, cloud computing is being used appropriately according to the purpose of use such as software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS).

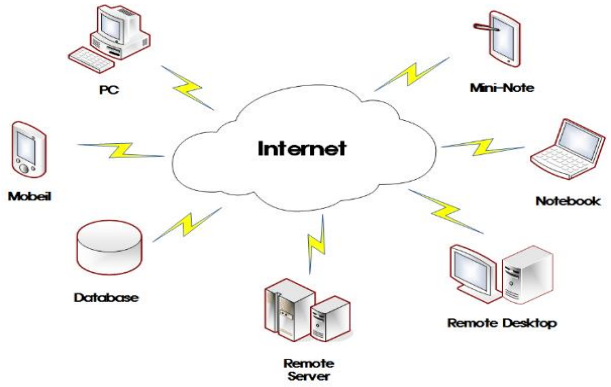


Fig. 1: Components of a Cloud Computing Environment.

Cloud computing has been improved in terms of efficiency (resource management, etc.) and convenience (location, time, etc.) rather than the existing Internet environment, but it is exposed to security threats in terms of the newly constructed cloud computing structure. Security threats in the cloud computing environment are not only security threats in the existing Internet environment but also security threats in the cloud computing environment (security threats during network transmission, security threats by administrators, security threats by virtualization engine hypervisors, etc.) And it is necessary to take measures accordingly [7].
Here are some detailed security practices for cloud computing security threats. In the case of security threats by the virtualization engine hypervisor, there is a threat of infecting the hypervisor's malware when running multiple virtual machines simultaneously in a cloud computing environment. In this case, there is a possibility that the malicious code is spread by the virtual machines, and there is a high possibility that another application (including the hypervisor) is hacked by including the hacking tool in the application program running in the virtual machine.
In the case of security threats by the administrator, there is no way for the user to check whether the user information stored or used in the cloud computing environment is illegally copied / moved / modified. In addition, there is no way for the administrator who manages the cloud computing to identify the user even if the user's personal information is leaked. In such a case, the separation of information ownership and information management in the cloud computing environment is not clear, so there is a situation where the responsibility is unclear if information is leaked or lost.
In the case of security threats during the network transmission process, physical resources are generated due to the sharing and centralization of physical resources in the cloud computing environment, and there is a threat that all services of the user using the cloud computing service may be stopped. Since cloud computing allows access to various terminals such as PC, smart phone, and smart TV, there is a possibility that not only the security threat of each terminal but also the user's information may be leaked if the mobile terminal is lost.

2.2. Federated identity management system

Most of them use a federated identity management system to protect user information in a cloud computing environment. There are Liberty Alliance, Kerveros, TCG (Trusted Computing Group), Windows Live ID, and OPENID [8-12].
The Liverty alliance was created to provide services to users after more than 140 companies have joined together to build cloud computing [8]. However, the Liverty alliance is vulnerable to man-in-the-middle attacks on client and proxy profiles. The problem is that the awareness provider used by the Liverty Alliance knows the identifiers of all users who are using the cloud computing service. Therefore, the administrator providing the service compromises with the recognition provider in order to connect the anonymity of the user, thereby inducing the result that is vulnerable to the man-in-the-middle attack. To solve this problem, the service provider must be able to connect to a single sign-on identifier based on a possible user network address while providing cloud computing.
Coverrus provides cloud computing services based on infrastructure [9]. The coverers share a long-term secret key with the authentication server to prevent the service provider from pre- venting the user's privacy risks. However, because it requires the user's trusted path to the password, it is vulnerable to man-in-the-middle attacks, such as password guessing, and internal attacks such as Trojan horses.
TCG (Trusted Computing Group) provides a service that can use hardware and software in a cloud computing environment [10]. The TCG shall initialize the Trusted Platform Module (TPM) so that the attributes of the hardware devices and software running in the distributed environment are reliable. Because TCG uses more trusted hardware than other federated identity management systems, TCG has the advantage of improving the security and privacy of the elements that make up the cloud environment. However, TCG is a disadvantage when the credibility of elements constituting a cloud environment is lowered, the security threat is higher than other systems (Liberty Alliance and Kerveros).
Windows Live ID is a federated identity management system offered by Microsoft to provide cloud computing services [11]. However, Windows Live ID has the advantage of making it easier to share documents, photos, and other files in Windows packages and to switch from one Windows Live ID to another in Internet Explorer, but only in the constrained environment of the Windows environment Therefore, it has a disadvantage that it can not receive services other than the Windows environment.
OpenID is an authentication method managed by the nonprofit Foundation OpenID Foundation [12]. OpenID is a benefit that Internet users do not need to create and manage new accounts every time they visit every site. However, OpenID is provided by an identity provider (or shortly idP, sometimes an i-broker) and has the disadvantage of authenticating only at one site they trust. OPENID has a simple identity hierarchy based on the OAuth 2.0 protocol. OAuth 2.0 defines a mechanism for getting and using access tokens to access protected resources, but it does not define standard methods that provide identity information.

2.3. Previous research

In this paper, we propose a new cloud management system for cloud computing, which is based on cloud computing. In particular, research on key generation and sharing of mobile devices operating in a cloud computing environment is one of the important factors in resolving cloud computing security[13-18].

The V. Shoup et. al technique is one of the extensions of the M. Bellare et. al model [14,15]. This technique uses the three keys used in the M. Bellare et. al model to improve safety. In addition, this scheme has advantages that the secret key used in the smart card is used longer than the existing scheme and is not compromised from the third party[15]. However, there is a disadvantage that when one of two objects using a secret key is compromised, security is easily exploited.

E. Liao et. al scheme integrates the password and attributes of smart cards used in the cloud environment [13]. The most common difference is that the password used in this technique is the number of passwords used for authentication rather than the password content. In this technique, authentication is performed by integrating the number of passwords and attributes because the client-server architecture used in the cloud environment requires stronger authentication than the existing architecture.

S. Lee et al. method proposed a public key generation method to authenticate a device in a cloud computing environment and a public key authentication method when a device moves to another cloud server [16]. This technique has the advantage of improving the authentication efficiency of the mobile device, but it has a disadvantage that the identifier, the password, and the PKI used in the authentication are transmitted to the plain text, not the cipher text, so that the data can be easily stolen from the attacker.

Z. Shen et. al scheme proposed a system to efficiently provide services supported by cloud computing. This model is characterized by a theoretical prototype combined with a trusted platform [17]. The A. Celesti et. al model proposed a management reference structure for the correct use of the identifiers used in cloud computing [18].

In recent research, we are studying whether the specific resources used in the cloud computing environment can be shared and used by mobile devices moving to other cloud environments. These studies, however, have the problem that the overall efficiency of the cloud environment drops when frequent use of a certain resource by dozens of users [19-24].

3. Efficient key distribution protocol for cloud environments

This section proposes an efficient protocol that can securely distribute keys to mobile devices in a cloud environment. The biggest difference between the proposed protocol and the existing protocol is that it does not use additional cryptographic algorithms.

3.1. Overview

In this paper, we propose a one-way hash function and XOR operation for the user information of the user, so that all users can share the resource without interruption to the physical resource, the key distribution and authentication can be performed.

The proposed protocol works in the system structure as shown in Fig. 2 in order to guarantee anonymity without unnecessary information disclosure of the mobile device when sharing the authentication information between the authentication server and the mobile device. As shown in Fig. 2, the system configuration for the proposed protocol is composed of one authentication server performing a master role and a plurality of mobile devices performing a slave role.

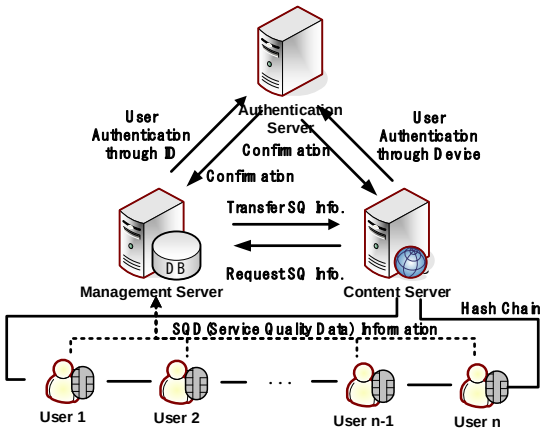


Fig. 2: Authentication Management Structure of Proposed Protocol.

When the authentication server attempts to read and store the authentication information of the mobile device on the distributed file system, the authentication server inquires the location of the block stored in the mobile device or determines the mobile device to store a copy of the authentication information. The intermediate device (management server, content server, etc.) existing between the authentication server and the mobile device manages the data input / output request of the authentication server. The authentication server verifies the normal operation of the mobile device and the block list in the mobile device through the block report, and utilizes it when the authentication server requests to read and store the file. In order to improve the efficiency of the initial authentication and management, the proposed protocol uses authentication that can manage data integration management information between the authentication server and the mobile device so that normal authentication and identification can be performed.

3.1.1. Create a federated identity for device authentication

When the authentication server requests the authentication information of the mobile device, the authentication server registers the anonymous ID, the mobile device information (device number, authority class, random value, etc.) of the mobile device, the time stamp, and the like in advance in the authentication server. By combining the information of the mobile devices registered in the authentication server and generating the federated ID that can be used in the cloud environment, the authentication server can use the federated ID to identify and authenticate the mobile device. In order to generate the federated ID, the information of the mobile device registered in advance in the authentication server is combined.

The generation of the federation ID is performed by combining the anonymous ID of the authentication server and the mobile device, and a federation ID that can be used in the cloud environment is generated by applying the authority level and time stamp of the generated authentication server. The random number in the anonymous ID is for the authentication server to respond to an attack such as a replay attack and the authority level to restrict access to the authentication information of the mobile device according to the authorization level of the mobile device. The time stamp is information for the valid period of the signature among the information used by the authentication server to access the authentication information of the mobile device.

3.1.2. Attack model

In the proposed protocol, if an attacker attempts to illegally access the authentication information of the mobile device, it uses an attack model that can not retrieve authentication information from the mobile device without inducing a fail-open state or modifying information. However, the proposed protocol assumes that the attacker is able to retrieve the data of the mobile device without modifying the authentication information of the mobile device or to access it through the program (or hacking program). If the attacker

can collect authentication information from the mobile device regardless of whether the authentication server exists or not, the security attack is successful. In the proposed attack model, disruptive attacks such as DoS are excluded. The proposed model also assumes that the attacker can not physically measure the real-time propagation signal of the patient without detection. This is because the cloud environment is safe and suits scenarios in which an attacker's presence can be minimized. We distinguish the attacker's attack strategy only in two respects. First, when an attacker tries to masquerade as a manager by leaking a shared key between a mobile device and an authentication server through a brute force search of a mobile device or a past authentication record. Second, an attacker can not prevent the mobile device from being safe It is assumed that the authentication server can be deceived by selectively jamming the authentication information from the mobile device.

3.2. Notations

Table 1 summarizes the terms used in the proposed protocol.

Table 1: Notations

Parameter	Notation
P_i	The prime number assigned to the i-th device
P	A prime generated by$\prod_{i=1}^{n}(p_{i+1}-p_i)$
g	Primitive element
$\mathbb{Z}_P$	The integer set ofP
X	Secret key group information of the mobile device
X'	Group secret information on the authentication server
y	Secret key secret information of mobile device
y'	Authentication server secret information
r	Random number inset $_R\mathbb{Z}_P$
t	Timestamp
$\|$	XOR operation

3.3. Key distribution protocol

In the key distribution protocol step, the manager uses the discrete logarithm problem to obtain the largest prime number P that has not been disclosed as $\prod_{i=1}^{n}(p_{i+1}-p_i)$. Where n denotes the number of mobile devices and $P_i$denotes the prime number of the i- th mobile device.

$$P = \prod_{i=1}^{n}(p_{i+1}-p_i) \quad (1)$$

The authentication server selects the primitive element g on $\mathbb{Z}_P$ and then discloses it to the mobile device.
The mobile device generates secret key k_i of the mobile device as Equation (2) to select secret information X of the mobile device and notifies the authentication server of Equation (3). Where n is the number of mobile devices.

$$X = \sum_{i=1}^{n} k_i \quad (2)$$

$$y \equiv g^X \bmod P \quad (3)$$

The mobile device computes T by selecting a random number r$\in$ R$\mathbb{Z}$Pand a time stamp t, and then transmits r, t, and T to the authentication server and transmits the information T for synchronizing with the information of the mobile device to the authentication server Deliver the request.

$$T \equiv (g^{r||t})X \bmod P \quad (4)$$

The authentication server computes Eq. (5) using the information of the received formula (4), confirms the mobile device, and transmits y, g^r, r to the authentication server.

$$T \equiv y^{r||t} \bmod P \quad (5)$$

The authentication server decides the session key SK$\in$ R$\mathbb{Z}$P and computesr', T', T'' asr', T', T'', T''' are stored. Where X' is the secret information of the authentication server.

$$r' = r + X' \bmod P\text{-}1 \quad (6)$$

$$T' \equiv (g^r)^{X'} \bmod P \quad (7)$$

$$T'' \equiv (y^{r'} \bmod P) \oplus SK \quad (8)$$

The authentication server calculates Eq. (9) and confirms the mobile device and sends Eq. (10) to the authentication server.

$$T' \equiv (y')^r \bmod P \quad (9)$$

$$T'' \equiv (y' \cdot g^{r'}) \bmod P \quad (10)$$

The mobile device computes $(T'')^X$ mod P$\oplus T' \equiv$SK from T'received from the authentication server and uses the session key SK.

3.4. Device-to-device mutual authentication protocol

In the mutual authentication protocol process, the server uses the session key SK generated by the key distribution protocol in advance for the mobile devices connected to the database and requiring mutual authentication. The session key SK does not provide any information form other than the authentication server to the third party. Also, the session key SK is used to encrypt / decrypt key information between the server and the mobile devices. In this process, the authentication server and the mobile device divide their random number r $\in$ RZP into arbitrary sizes and use them for mutual authentication. This process does not require additional cryptographic computation for the authentication server and the mobile device, which is advantageous in that the calculation cost is low. The detailed operation of the mutual authentication process between devices is as follows.

- Step 1: The authentication server requests the secret information X of the mobile device using the session key SK previously given to the mobile device to receive the service in the cloud environment. At the end of the request, the authentication server waits for a response from the mobile device. At this time, the secret information X of the mobile device that has been responded is a value that was initially registered in the database of the authentication server as a value selected using the secret key (k_i) of the mobile device. The secret information X of the mobile device is used to protect the random number $r_S \in$ R$\mathbb{Z}$Pgenerated by the authentication server. The authentication server randomly generates a security identifier SID corresponding to secret information X of the mobile device.

Step 2: The authentication server applies secret information X of the mobile device received from the mobile device to g^Xmod P to generate secret key secret information y of the mobile device. The generated secret information y is used in mutual authentication between the authentication server and the mobile device in combination with the identifier $ID_M$of the mobile device (C $\equiv$ y$\oplus ID_M$).

- Step 3: The authentication server encrypts the secret information X and C ($\equiv$ y$\oplus ID_M$) of the mobile device using the session key SK as E_{SK} (X, C). The C value obtained by combining the secret information y and the identifier ID_M of the mobile device. It is substituted into the hash function (=h_C (SID)) and transmitted to the mobile device through the challenge.
- Step 4: The mobile device extracts the hash value(= h_C (SID|| X))) received from the authentication server by using the C value obtained by combining the secret information y and the identifier ID_M of the mobile device Obtain the identifier SID'. This process can prevent spoofing attempts to il-

legally exploit mobile devices. The mobile device concatenates the hash value received from the authentication server with the security identifier SID of the mobile device and the secret information $X'$of the mobile device, and assigns the value(=$h_{k_i}(h_C\ (SID'||X')\|$ SID $\|X')$) to the hash function using the secret key (k_i). The mobile device replaces the generated value (=$h_{k_i}(h_C\ (SID'||X')\|$ SID $\|X')$)을 with S to the hospital(leader).

- Step 5: The authentication server uses the secret key (k_i) of the mobile device registered in the authentication server to hash S to check whether synchronization with the mobility entity is performed using the information received from the mobile device, and then h_C (($SID'||X')\|$ SID $\|X'$). The authentication server compares the security identifier SID' and secret information X' extracted from h_C ($SID'||X'$) with the previously registered security identifier SID and secret information X.
- Step6 : The authentication server compares the secret information X selected using the secret key (k_i) of the mobile device registered in advance with the secret information X' received from the mobile device. If the two secrets match, the mobile node transmits C, which is a combination of the mobile device information, to the database using the shared key SK_{S-D} shared between the authentication server and the database for resynchronization. If they do not match, it detects that asynchronization has occurred during transmission and retransmits the synchronization request message and h_C ($SID'||X'$) to the patient for resynchronization.
- Step 7: The database searches the secret information $X'$of the mobile device and compares it with the secret information X of the mobile device registered in advance. Then, the database generates a session key SK corresponding to the secret information X of the registered mobile device and transmits it to the authentication server. If the information corresponding to the session key SK held by the authentication server is transmitted, the database newly updates the secret information X' of the transferred mobile device and the communication connection state information I, and then transmits the information to the authentication server. And terminates.
- Step 8: The authentication server hashes the secret information $X'$of the newly updated mobile device and the communication connection state information I as $h_{SK}(X'\|\mathrm{I})$, and then transmits the hash to the mobile device.
- Step 9: The mobile device extracts $h_{SK}(X'\|\mathrm{I})$ received from the authentication server using the session key SK. And updates the secret information $X''$of the mobile device and the communication connection state information I. At this time, the mobile device compares the secret information $X'$of the mobile device derived from $h_{SK}(X'\|\mathrm{I})$ with the secret information X of the secret information mobile device of the existing mobile device, Send a confirmation message to the user and exit.
- Step 10: The authentication server transmits an OK confirmation message to the database if mutual authentication with the mobile device is normally performed.

4. Evaluation

The evaluation of the proposed protocol performs the security evaluation and performance evaluation to distribute and authenticate the key so that the mobile device can efficiently use the physical resources in the cloud environment. The performance evaluation evaluates the authentication processing time, the authentication processing rate of the server per unit time, the communication delay time between the authentication server and the mobile device, and the authentication overhead of the server.

4.1. Security evaluation

In order to prevent the reuse attacks using mobile devices in the cloud environment, the proposed protocol uses the largest prime number P (= $\prod_{i=1}^{n}(p_{i+1}-p_i)$) for synchronizing information X with the information of the mobile device, and notifies the authentication server every time the session key SK is transmitted to the third party, Even if it is eavesdropped. In the proposed protocol, information T and $(T'')^X$ mod P⊕T' received from the authentication server is computed and it is safe for reuse attack because the mobile device calculates and uses the session key SK(≡ $(T'')^X$ mod P⊕T') with information T'received from authentication server.

In the proposed protocol, when a third party tries to generate r'using the serial number r of the mobile device, the third party is safe from spoofing attacks because it does not know the secret key (k_i) of the mobile device. Also, when the third party transmits the output value T generated by the mobile device to the authentication server, it eavesdrops on the previous session and transmits the stored value instead. However, with the T transmitted to the authentication server, the authentication server determines the session key SK∈ RℤP and compares and compares r', T', T''.

Since the proposed protocol uses the random number (r , r') generated by the mobile device and the authentication server each time the mobile device accesses the authentication server, the information T', T'', T''' are exposed, it does not recognize the secret information $X'$of the authentication server. The authentication server uses the session key SK generated by the authentication server so that the third party can not illegally use the information of the mobile device. Table 2 compares the proposed protocol with existing protocols by user authentication, authorization, biometric authentication, message authentication, user convenience, and efficiency.

Table 2: Comparative Analysis of Proposed Protocol and Existing Protocol

Division	User Authentication	Authority authentication	Bio-authentication	Message Authentication	User convenience	Efficiency
Ipath [19]	ID/PasswordBased	Access control policy	-	impossible	usually	usually
OpenEMed [20]	Certificate-based	Access control policy	-	possible	usually	usually
TeleCardio-FBC [21]	Public key based	-	-	impossible	usually	usually
WBASN [22]	ID/PasswordBased	-	Use	impossible	Convenient	usually
CodeBlue [23]	Public key based	-	Use	possible	Convenient	usually
Medintegra Web [24]	ID/PasswordBased	Access control policy	use	impossible	Convenient	usually
Proposed protocol	One-way hash function and XOR operation	Access control policy	-	possible	Convenient	High

In Table 2, existing protocols provide privacy and authentication of mobile devices as a basis. In addition, ID, password, and certificate necessary for authentication are used for public key base and biometric authentication. However, since the proposed protocol combines the one-way hash function and the XOR operation based on

the discrete logarithm problem in order to authenticate the mobile device in the cloud environment, the computational efficiency is higher than that of the existing protocol. In addition, if the existing protocol has lost the ID, the password and the certificate, the authentication can be performed even if the user does not exist. Therefore, there is a problem that the integrity of the data used in the cloud computing is not guaranteed. However, It is more secure than existing systems because it does not need to be lost.

4.2. Performance evaluation

4.2.1 Authentication process time

Table 3 shows the estimated authentication processing time when the time required for the mobile device operating in the cloud environment to access the authentication server and performing the authentication is applied to the proposed protocol for each encryption algorithm. The encryption algorithms used in Table 3 are HMAC (SHA-1), RIPEMD-256, AES / ECB (256-bit key) and RC5.

Table 3: Authentication Processing Time of Authentication Server by Encryption Algorithm

Unit : ms

Cryptographic algorithm Number of mobile devices	HMAC (SHA-1)	RIPEMD-256	AES/ECB (256-bit key)	RC5 (r=8)
100	33.7	38.5	27.6	29.3
250	38.3	44.9	29.9	34.2
500	48.2	51.2	40.7	45.3
1,000	55.6	61.7	49.5	52.7

Figure 3 shows the processing time required to authenticate the mobile device by comparing and analyzing important information of the mobile device with information pre-stored in the authentication server when the mobile device wants to receive the service from the authentication server in the cloud environment. As shown in Fig. 3, the proposed protocol increases the authentication processing time of the server by 4.1% compared with the existing protocol due to the increase of the number of mobile devices depending on the main information (kind, function, characteristic and attribute) required for authentication and the subnet configuration of cloud computing. Results were obtained. The result shown in Fig. 4 is that when the proposed protocol requests authentication to the authentication server, subnets are classified according to the main information of the mobile device, and since the hash function based on the discrete logarithm is combined with the XOR operation, This is the result. In addition, when the mobile device tries to request authentication to the server existing in the cloud environment, the authentication process is performed according to the access policy of the authentication server. Therefore, even if a large amount of mobile devices request authentication, The search processing time is not increased.

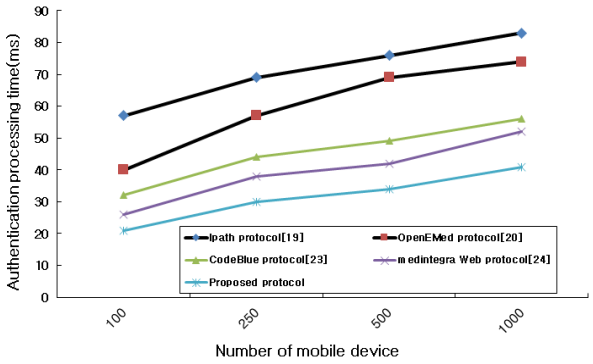


Fig. 3: Authentication Processing Time of Server.

4.2.2. Authentication throughput of server per unit time

Figure 4 compares the authentication throughput of the authentication information processed per unit of time in the authentication server with the existing scheme when the mobile device requests authentication in the cloud environment. As shown in Fig. 4, when the authentication server performs the authentication process requested by the mobile device, the proposed protocol generates the secret key (k_i) of the mobile device, the secret information X of the mobile device, the random number r $\in_R$ ℤP generated by the mobile device, since the mobile device is configured as a hierarchical subnet after selecting timestamp t, the average throughput of server per unit time is 6.5% higher than the existing protocol. The reason for this result is that the probability-based global attribute information is extracted from the mobile device so as to have a minimum delay time in verifying various authentication information of the mobile device requested to the authentication server. In order to easily identify the mobile device, authentication information is converted into polynomial form using discrete logarithm and stored in the server.

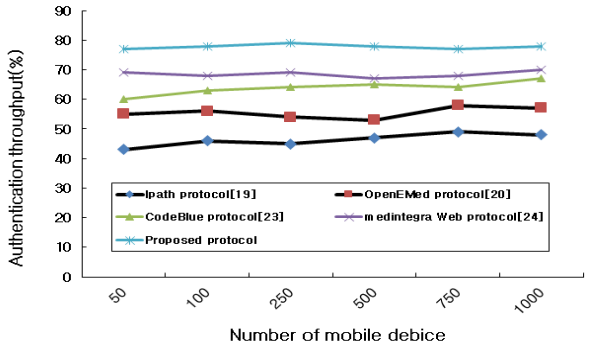


Fig. 4: Authentication Throughput of Server per Time.

4.2.3. Communication delay time between authentication server and mobile device

Figure 5 shows the communication delay time when the authentication server verifies important information of the mobile device in the cloud environment. In Fig. 5, the proposed protocol uses a polynomial to extract the authentication information of the mobile device from the server as a vector in order to authenticate the important information of the mobile device. Since this method is used, the proposed protocol has an average improvement of 9.3% over the existing protocol. In particular, because the proposed protocol combines the one - way hash function and the XOR operation based on the discrete logarithm problem to authenticate the mobile device, the communication delay between the authentication server and the mobile device is improved. In addition, the existing protocol uses ID, password, and certificate. In the proposed protocol, however, communication delay time is reduced because authentication is performed using only the main information of the mobile device.

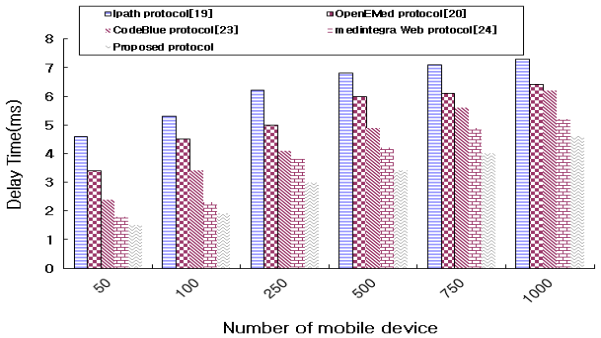


Fig. 5: Delay Time between Authentication Server and Mobile Device.

4.2.4. Authentication overhead of the server

Table 4 shows the overhead change of the authentication server according to the number of mobile devices when the authentication

information requested by the mobile device is processed by the authentication server. As shown in Table 4, the proposed protocol yields an average of 11.5% lower than the existing protocol.

Table 4: Authentication Overhead of Server

Protocol No mobile device	Ipath protocol [19]	OpenEMed protocol [20]	CodeBlue protocol [23]	Medintegr a Web protocol [24]	Proposed protocol
50	0.718	0.653	0.607	0.553	0.462
100	0.745	0.684	0.614	0.584	0.482
250	0.751	0.715	0.597	0.565	0.479
500	0.778	0.674	0.607	0.574	0.481
750	0.769	0.707	0.621	0.587	0.500
1,000	0.773	0.716	0.639	0.586	0.510

5. Conclusion

With the development of the Internet, the demand for cloud computing services is increasing. However, since important information of a mobile device used in a cloud environment can be leaked even if a server is stored after being encrypted, there is a need for a key generation and authentication necessary for data encryption / decryption. In this paper, we propose a protocol that can efficiently authenticate and distribute keys for mobile devices to securely encrypt and decrypt data in the cloud environment. The proposed protocol aims to maximize the efficiency and cost reduction of key generation, which can securely transmit and receive data, in a situation where the size of data used in the cloud environment and the storage location are increasing. In order to solve this problem, the proposed protocol uses the key information of the mobile device to generate the session key and use it for the encryption. Therefore, not only the security problem of the mobile device can be solved, but also the efficiency of the authentication server can be improved. As a result of the performance evaluation, the proposed method improved the authentication processing time by 4.1% on average compared with the existing protocol, and the average throughput rate of the server per unit time was 6.5%. In addition, the communication delay time between the authentication server and the mobile device improved by 9.3% on average, and the authentication overhead of the server was 11.5% lower than that of the conventional method. In future studies, we will apply the proposed protocol to the actual environment based on the results of this study and compare it with the results obtained from the theoretical studies.

6. Acknowledgment

This paper has been supported by 2018 Hannam University Research Fund.
This work was supported by the Security Engineering Research Center granted by the Ministry of Trade, Industry and Energy.

References

[1] Matsunaga R,Ricarte I, Basso T, Moraes R, Towards an Ontology-Based Definition of Data Anonymization Policy for Cloud Computing and Big Data,47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), 22017, pp. 75-82.
[2] Banerjee A, Hasan M, Rahman A, Chapagain R, CLOAK: A Stream Cipher Based Encryption Protocol for Mobile CloudComputing, IEEE Access, 2017, PP(99), pp.1-1.
[3] Ye X,Yin Y, Lsan L, Energy-Efficient Many-Objective Virtual Machine Placement Optimization in a Cloud Computing Environment, IEEE Access, 2017, 5, pp. 16006-16020.
[4] Stergiou C, Psannis K E, Algorithms for Big Data in Advanced Communication Systems and CloudComputing, IEEE 19th Conference on Business Informatics (CBI), 2017, 01, pp. 196-201.
[5] Taleb T, Samdanis K, Mada B, Flinck H, Dutta S, Sabella D, On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration, IEEE Communications Surveys & Tutorials, 2017, 19(3), pp. 1657-1681.
[6] Shen P, Zhou Y, Chen K, Enhancing reliability via checkpointing in cloud computing systems, China Communications, 2017, 14(7), pp. 1-10.
[7] Li J, Yao W, Zhang Y, Qian H, Han J, Flexible and Fine-Grained Attribute-Based Data Storage in Cloud Computing, IEEE Transactions on Services Computing, 2017, 10(5), pp. 785-796.
[8] Krishnamurthi G, Chan T K, Using the Liberty Alliance Architecture to Secure IP-level Handovers, 1st International Conference on Communication Systems Software & Middleware, 2006, pp. 1-10.
[9] Al-Ayed F, Liu H, Synopsis of Security: Using Kerberos Method to Secure File Transfer Sessions, International Conference on Computational Science and Computational Intelligence (CSCI), 2016, pp. 1016-1020.
[10] Achemlal M, Gharout S,Gaber C, Trusted Platform Module as an Enabler for Security in Cloud Computing, Conference on Network and Information Systems Security, 2011, pp. 1-6.
[11] Yamansavascilar B, Guvensan M A, Yavuz A G, Karsligil M E, Application identification via network traffic classification, International Conference on Computing, Networking and Communications (ICNC), 2017, pp. 843-848.
[12] Alves J M, Rodrigues T G, Beserra D W, Fonseca J C, Endo P T, Kelner J, Multi-Factor Authentication with OpenId in Virtualized Environments, IEEE Latin America Transactions, 2017, 15(3), pp. 528-533.
[13] Liao I E, Lee C C, Hwang M S, A password authentication scheme over insecure networks, Journal of Computer. System Sciences, 2006, 72(4), pp. 727-740.
[14] Shoup V, Rubin A, Session key distribution using smartcards, Proceedings of the 15th annual international conference on Theory and application of cryptographic techniques, 1996, pp. 321-333.
[15] Bellare M, Rogaway P, Provably secure session key distribution-The third party case, Proceedings of the twenty-seventh annual ACM symposium on Theory of computing, 1995, pp. 57-66.
[16] Lee S, Ong I, Lim H T, Lee H J, Two factor authentication for cloud computing, International Journal of KIMICS, 2010, 8, pp. 427-432.
[17] Shen Z, Li L, Yan F, Wu X, Cloud Computing System Based on Trusted Computing Platform, International Conference on Intelligent Computation Technology and Automation (ICICTA),2010, 1, pp. 942-945.
[18] Celesti A, Tusa F, Villari M, Puliafito A, Security and Cloud Computing: InterCloud Identity Management Infrastructure, 19th IEEE International Workshop on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE), 2010, pp. 263-265.
[19] Vijay R S, Madhavi Y, Venkateswarlu C, Ipath: Path Inference in Wireless Sensor Networks, International Journal of Innovative Research in Computer and Communication Engineering, 2017, 5(3), pp. 4226-4232.
[20] OpenEMed, http://openemed.org/.
[21] Holzinger A, Sammeer P, Hofmann-Wellenhof Rainer, Mobile Computing in Medicine: Designing Mobile Questionnaires for Elderly and Partially Sighted People, International Conference on Computers for Handicapped Persons, 2006, pp. 732-739.
[22] Moteiv, www.moteiv.com, March 20 2006.
[23] Shnayder V, Chen B, Lorincz K, FulfordJones T R F, Welsh M, Sensor networks for medical care, Technical Report TR-08-05, Harvard University, Division of Engineering and Applied Sciences, 2005.
[24] Apollohospitals, http://www.apollohospitals.com, 2006.