Received: 28 May 2020 | Revised: 29 September 2020 | Accepted: 17 October 2020

DOI: 10.1002/ett.4202

SPECIAL ISSUE ARTICLE

WILEY

Public key encryption with equality test for Industrial Internet of Things system in cloud computing

Ganesh Gopal Deverajan[1] | **V. Muthukumaran**[2] | **Ching-Hsien Hsu**[3,4] | **Marimuthu Karuppiah**[5] | **Yeh-Ching Chung**[6] | **Ying-Huei Chen**[7]

[1]Department of Computer Science Engineering, Chandigarh University, Mohali, India

[2]Department of Mathematics, School of Applied sciences, REVA University, Bangalore, India

[3]Department of Computer Science and Information Engineering, Asia University, Taichung, Taiwan

[4]Department of Medical Research, China Medical University Hospital, China Medical University, Taiwan

[5]Department of Computer Science and Engineering, SRM Institute of Science and Technology, Delhi-NCR Campus, Ghaziabad, Uttar Pradesh, 201204, India

[6]School of Science and Engineering, The Chinese University of Hong Kong, Shenzeng, China

[7]College of Humanities and Social Sciences, Asia University, Taichung, Taiwan

Correspondence
Ganesh Gopal Deverajan, Department of Computer Science Engineering, Chandigarh University, Mohali, Punjab 140413, India.
Email: dganeshgopal@gmail.com
Ching-Hsien Hsu, Department of Computer Science and Information Engineering, Asia University, Wufeng, Taichung 41354, Taiwan Tel: +886-4233234566303
Email: robertchh@gmail.com

Funding information
GuangdongHong Kong-Macao Intelligent Micro-Nano Optoelectronic Technology Joint Laboratory, Grant/Award Number: 2020B1212030010; This work is partially supported by the National Natural Science Foundation of China, Grant/Award Number: 61872084

Abstract

Present day world have evolved from traditional environment to smart industries using IoT scheme which in turn forms Industrial Internet of Things (IIoT), which significantly elaborated by providing enhance integration using smart communication through IoT based sensors. IIoT has been providing cost reduction and enhancement in technology by bringing availability, flexibility and data sharing through real time scenario. Despite being unsecure environment of cloud, the privacy of data transfer and information confidentiality is guaranteed. In this context, this work presents a Public Key Encryption with Equality Test based on DLP with double decomposition problems over near-ring. Computation Diffie-Hellman is utilized in algebraic structure which involves DLP with Double Decomposition problem for proposing a Public Key Encryption with Equality Test which provides more security to the scheme. The proposed method is highly secure and it solves the problem of quantum algorithm attacks in IIoT systems. Further, the suggested system is significantly secure and it prevents the chosen-ciphertext attack in type-I rival and it is indistinguishable against the random oracle model for the type-II rival. The recommended scheme is highly secure and the security analysis measures are comparatively stronger than existing techniques. Search time of the proposed scheme is 150 milliseconds for which the number of attributes is 50 and when comparing to the decryption time of the proposed model which is lower when compared to other existing scheme for 50 attributes.

1 | INTRODUCTION

Internet of Things (IoT) is a processing idea portraying pervasive association with the Internet, turning regular articles into associated gadgets. The vital methodology in IoT based model is transmitting billions or even trillions of keen items skilled to detect the encompassing condition, communicate and transfer of detailed information, and afterward criticism to nature. It is expected that constantly in 2021 it is associated with 28 billion associated gadgets.[1] Associating unpredictable items to enhancing Internet, and security of ventures and society, and empower proficient communication between the physical world and its computerized partner, for example, what is generally tended to as a cyberphysical system (CPS). It is generally portrayed to be problematic innovation for understanding problems in present world, for example, savvy urban communities, shrewd transportation, contamination checking, associated human services, to name a couple. Typical structure of IIoT is illustrated in Figure 1. IIoT includes the machine to machine spaces, mechanical correspondence advancements with computerization applications. IIoT prepares to better comprehension of the assembling procedure, along these lines empowering proficient and practical creation.

Adaptability and versatility required by IoT interchanges are regularly tended to utilizing remote connections. Before, remote innovations in modern applications were for the most part in light of specially appointed arrangements, for example, exclusively created for interfacing moving parts or difficult to-arrive at gadgets. Recently, gauges deliberately intended for the business (eg, WirelessHART[2] and ISA100.11a[3]) were discharged. Be that as it may, they face restrictions regarding adaptability and inclusion at the point when exceptionally enormous regions should be secured. While cell advancements, for example, 3/4/5G advances guarantee to interface separations of gigantic gadgets, they require framework support and authorized band.[4] IIoT applications normally require moderately little throughput per hub and the limit is certainly not a principle concern. Rather, the need of interfacing a very huge number of gadgets to the Internet requiring little to no effort, with restricted equipment abilities and vitality assets (eg, little batteries) make dormancy, vitality productivity, cost, dependability, and security/protection progressively wanted highlights.[5-9]

Meeting the previously mentioned necessities represents a number of key difficulties on the advancement of IIoT. Tending to these challenges is basic so as to guarantee a gigantic turn out of IIoT advancements. In this article, we explain the ideas of IoT, IIoT, and the present pattern of robotization and information trade in assembling advancements called Industry 4.0. We feature the open doors got by IIoT just as the difficulties for its acknowledgment. Specifically, we center around the difficulties related with the need of vitality productivity, continuous execution, conjunction, privacy, operational consistency and protection issues. We likewise give a deliberate outline of the best in class investigate endeavors and enhancement of research in near future bearings face the challenges of Industrial IoT. This is regularly utilized with regards to Industry 4.0, the Industrial Internet and related activities over the globe.

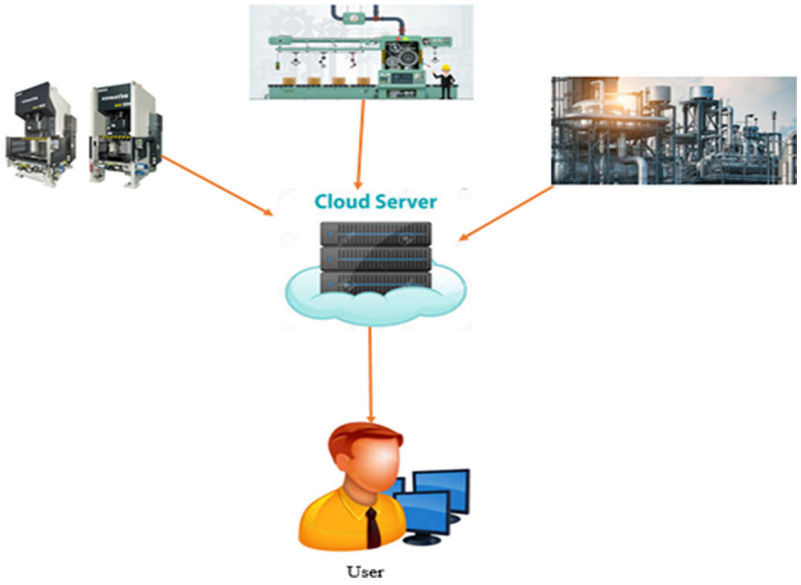


FIGURE 1 Typical structure IIoT systems

Industry 4.0 depicts another modern upset with an attention on mechanization, development, information, digital physical frameworks, procedures, and individuals.[10] With Industry 4.0, the fourth modern upheaval is determined to consolidating computerization and data spaces into the mechanical Web of things, administrations, and individuals. The correspondence foundation of Industry 4.0 enables gadgets to be available in obstruction freeway in the mechanical Internet of things, without yielding the uprightness of wellbeing and security.[11] The expression "mechanical Internet" was begat by Industrial mammoth GE to portray modern change in the associated setting of machines, digital physical frameworks, progressed examination, AI, individuals, cloud, and so on. GE and the Industrial Internet Consortium (IIC) chose that IIoT was an equivalent word for the Industrial Internet. IIoT is ready to carry uncommon chances to business furthermore, society. Associations like IIC and IEEE are striving to characterize and build up the IIoT. Every complete IoT network are similar in that they signify the association of four distinct parameters such as devices and sensors which is utilized for connectivity, processing of data and user interface for creating the manipulation of data once the data are stored in the cloud, processing of data is carried out using some software and based on the actions performed.

Thus from the above examples, it is clearly revealed that security remains to be a major problem across the IIoT environment. Especially, authorization remains to be a major issue. This research contributes the methodology through which attacks on IIOT systems can be circumvented. The major focus of this proposed model is to present a Public Key Encryption with Equality Test based on DLP with double decomposition problems over near-ring. This research findings is significantly effective in avoiding quantum algorithm attacks in IIOT based networks.

1.1 | Contributions

In order to solve this issue in this article, we propose a title for the IIoT environment. The proposed method is highly secure as it solves computation ciphertext attack based on Computational Diffie-Hellman problem in hard type-I adversary model and chosen-ciphertext attacks using decisional Diffie-Hellman problem for hard type-II adversary model. In this suggested IIoT scheme, the authorization mechanism is versatile such that four types of authorizations as shown in Figure 2 and it can be used to delegate servers in the cloud to execute the search functionality Major contributions of the article are listed as follows:

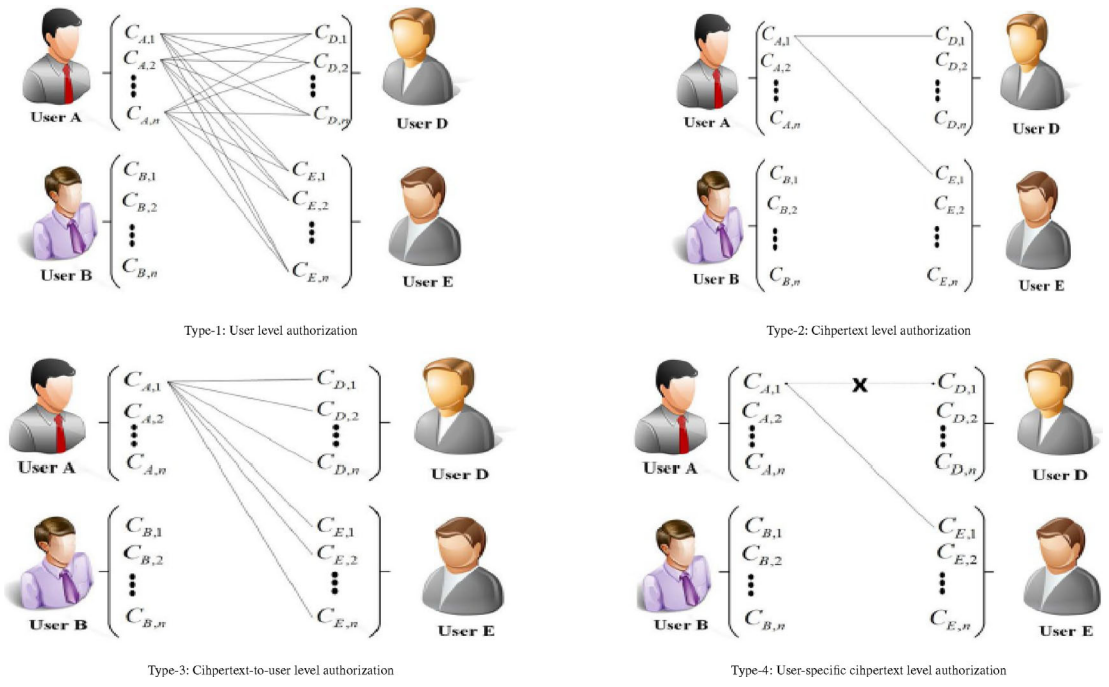


FIGURE 2 Four types of authorizations for IIoT

1. The proposed system is resistant against one way-chosen ciphertext (OW-CCA) attack in the case of $CDH^N_{a,b}$ in hard random oracle model for type-I adversary.
2. The proposed system is resistant against one way-chosen ciphertext (OW-CCA) attack in the case of $DDH^N_{a,b}$ in hard random oracle model for type-II adversary.

1.2 | Organization

The rest of the article is organized as follows: Section 2 provides clear description to some of the works related to the proposed scheme. Preliminaries and security models are defined in Section 3. A brief description to the proposed scheme is given in Section 4. Sections 5 and 6 define security and performance analysis. Section 7 concludes the article.

2 | RELATED WORKS

In last few decades, with advancement in cloud computing and tremendous evolution of data, there is an enormous consideration for storage technology in cloud. Encryption of data is crucial to make sure data which is sensitive is provided with privacy. To attain the aim of finding ciphertext without illuminating any plaintext information, Boneh et al[12] developed the primary PEKS (public key encryption with keyword search) scheme, where bilinear map is considered. Whereas, here in this scheme, complexity of searching becomes linear with number of keywords which are encrypted in each document. In calculation, trapdoors are transmitted with the help of secure channel. To overcome this problem, Baek et al[13] came with a protected channel free PEKS scheme (SCF-PEKS). In Reference 13, transmission of trapdoor can be done via a public channel with the help of public or private keys in could server. Rhee et al[14] pointed that the capabilities of attackers in the scheme security model is restricted. Scheme security model is strengthened,[13] and develop a PEKS scheme in the superior model. Byun et al[15] came across the fact that the latest PEKS scheme are prone to an offline keyword predicting attack due to the fact that the keywords are basically designated from the minor space than passwords and users are feared of using some particular repeatedly used keywords for searching a document. In order to overcome this problem, Rhee et al[16] suggested a PEKS scheme which includes tester which is designated.

Ma et al[17] proposed an encryption which has public key and which include eminence test scheme associated with flexible authorization. Fang et al[18] developed a SCF-PEKS scheme, which is efficient for keyword prediction attack below standard model. Whereas, abovementioned schemes lack from problem which has key management problem or key escrow issues. To overcome this issue, Peng et al[19] primarily brought the concept of certificateless public key encryption with keyword search (CLPEKS). Far ahead, Wu et al[20] illustrates that Peng et al scheme lacks from an keyword guessing attack which is off-line. In recent times, Ma et al developed two various CLPEKS strategies in References 21, 22, correspondingly. The mentioned scheme is also prone to attack which is of keyword guessing type hosted by malevolent structure insider. To alleviate IKGA, Xu et al[23] develop a Public-Key Encryption with Fuzzy Keyword Search plot (PEFKS), in which every catchphrase relates to a definite watchword search trapdoor and a fluffy catchphrase search trapdoor. As of late, Chen et al[24] present a double server PEKS plan to avoid IKGA. Huang et al[25] present the idea of Public Key Authenticated Encryption with Keyword Search (PAEKS), where the information proprietor scrambles every catchphrase, yet in addition validates it.

It is observed from the literature that securities of the above discuss PKEwET models works on the basis of assumptions relating to discrete logarithm problem (DLP). In Reference 20 Shora's quantum algorithm provides an effective measure to solve DLP at the polynomial time. As a result of this scheme, the existing approaches become highly insecure with the evolution of quantum computing techniques. Thus, it is mandatory to analyze new PKEwET schemes against quantum algorithm attack. In Reference 21 Magyarik and Wagner provide a cryptographic model based on noncommutative algebraic structure.

It is clearly envisioned from the literature[5,26-30] there exists several security and privacy protection measures for IIOT systems but it fails to provide efficient authentication and authorization facilities. Further, with secure authorization and authentication techniques complexity remains to be the major issue. In this context, the proposed method provides secured solution for IIOT systems with improved user authorization and complexity measures. Baza et al proposed a B-ride which is a privacy conservation scheme that allows fair payment along with trust atop blockchain network.[31]

Sakhnini et al mentioned the aspects involved in the security concerning to the internet of things aided smart grids.[32] Xu et al utilized the internet of things for different smart application in the field of manufacturing sectors for resource assignments in industrial sectors.[33] Zhou et al utilized keyword search using public key encryption in cloud environment. This work focuses on cloud computing popularization, various business and individual preferences.[34] Dwivedi et al presented some vital benefits and various other practices that are to be monitored in case of blockchain oriented security scenarios in IoT environment.[35] Thirumalai et al proposed an efficient scheme based on the public key for IoT and cloud based security.[29]

3 | MATHEMATICAL BACKGROUND

3.1 | Complexity assumptions

Discrete logarithmic problem with double decomposition problem (DDLPDCP):

Let $(N, +, .)$ be a noncommutative near-ring. Let a, b, ω, y be arbitrary elements of N and α, β, a be a random elements Z_p^*. The given $a, b, \omega \in N$ such that $y = a^{\alpha}\omega^{a}b^{\beta}$. Find $a, b \in N$ and $\alpha, \beta, a \in Z_p^*$. The process of DDLPDCP is given in Algorithm 1.

Thus the DDLPDCP is the hybrid of the DLP and the DDP. The complexity of the DDLPDCP and level of security of cryptosystem constructed on the DDLPDCP will differ on size of N and p. Therefore, to attain the security of order 2128 we may select a prime q of size approximately 25. The size of the prime p must be taken more than or equal to 48 bits.

Computational Diffie-Hellman problem ($CDH^{N}_{a,b}$):

Given a near-ring N and a quintuple $(a, b, a^{\alpha_1}\omega^{a1}b^{\beta_1}, a^{\alpha_2}\omega^{b1}b^{\beta_2}) \in N$ where $\alpha_1, \alpha_2, \beta_1, \beta_2 a_1, b_1 \in Z_p^*$, the objective is to compute $a^{\alpha_1+\alpha_2}\omega^{a_1+b_1}b^{\beta_1+\beta_2}$. Here $a, b \in N$, satisfied $\langle a\rangle \cap \langle b\rangle = \langle e\rangle$ and $ab \neq ba$.

Decisional Diffie-Hellman problem ($DDH^{N}_{a,b}$):

Given a near-ring N and a quintuple $(a, b, a^{\alpha_1}\omega^{a_1+b_1}b^{\beta_1}, a^{\alpha_2}\omega^{a_2+b_2}b^{\beta_2}, a^{\alpha_3}\omega^{a_3+b_3}b^{\beta_3}) \in N$ where $\alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3, \in Z_p^*$, the objective is to decide whether $a^{\alpha_3}\omega b^{\beta_3} = a^{\alpha_1+\alpha_2}\omega b^{\beta_1+\beta_2}$. Here $a, b \in N$, satisfied $\langle a\rangle \cap \langle b\rangle = \langle e\rangle$ and $ab \neq ba$

Computational Diffie-Hellman problem ($CDH^{N}_{a,b}$) for trapdoor function:

Indeed, it allows to make chosen ciphertext secure schemes from any one-way encryption scheme: for any one-way function, if a trapdoor allows to get back a part of the preimage, one can base a chosen-ciphertext secure encryption scheme on the relying computational problem. More concretely, from any one-way encryption scheme (which is the weakest requirement one can make about an encryption scheme) and just two more hashing, one can make a highly secure cryptosystem relying only on the same assumption as the one-wayness of the original scheme, which is generally

Algorithm 1. Discrete logarithmic problem with double decomposition problem

Input: a, b, $\omega \epsilon N$ such that $y = a^{\alpha}\omega^{a}b^{\beta}$
Output: Secret parameters $\alpha, \beta, a \epsilon Z_p^*$

1: **for** $i \leftarrow 1$ to p-1 **do**
2: $\quad \overline{\omega} \leftarrow \omega^{a}$
3: **end for**
4: **for** $j \leftarrow 1$ to μ **do**
5: $\quad y_j \leftarrow a^{\alpha}\overline{\omega}b^{\beta}$
6: $\quad$ Compare $y = y_j$
7: **end for**
8: **if** $y = y_j$ **thenreturn** (a^{α}, i) & exit
9: **else**
10: $\quad$ go to next step
11: **end if**
12: $j \leftarrow j + 1$
13: $i \leftarrow i + 1$
14: End

a really difficult computational problem (at least more difficult than just decisional ones). We then apply this generic transformation to many well-known one-way functions to provide the best schemes of their families: the most efficient scheme based on the computational Diffie-Hellman problem.

3.2 | Security models

In this subsection, we review the security model of PKEwET-FA, which was defined in.[19]

The security model includes six algorithms: Setup, KeyGen, Encrypt, Decrypt, Authorization, and Test. Suppose that the system has a unique index for user. The Setup algorithm establishes system parameters. The KeyGen algorithm generates the public key and private key for user. The Encrypt algorithm outputs a ciphertext for a message and the public key. The Decrypt algorithm outputs a message or using a private key. The Authorization algorithm generates the trapdoor with the private key. The Test algorithm takes two ciphertexts, the trapdoors as inputs and outputs 1 when they are the same message or 0 otherwise. Because the Type-4 authorization is a combination of Type-1 and Type-2 authorization, we leave out Type-4 authorization queries for simplicity and allow only Type authorization queries to the adversary in the security games.

The two types of adversaries as follows:

1. Type-I adversary: For type-I adversary, he/she cannot obtain any information except for the test information with type trapdoor information.
2. Type-II adversary: For type-II adversary, he/she cannot judge whether the challenge ciphertext is encrypted by which message without Type trapdoor information.

First we define OW-CCA security for Type authorization against type-I adversary as shown in Table 1.

In the security models of the type-I adversary, the adversary A_1 submits a target K which he/she wants to challenge before the game.

Here, $O_1(i) = KeyGen(i, S_p), O_2(i, T_i) = dec(Sk_i, CT_i),\ O_3(i, .) = ET - Auth(Sk_i, .), O_4 = O_1(i)$ or $O_4 = \perp$, and the condition is that $i \neq K$, $O_5(j, CT_j) = O_2(j, CT_j)$ or $O_5(j, CT_j) = \perp$, and the condition is $CT_j \neq CT^*$, $O_6 = O_3$

Second we define IND-CC a security for type $-\beta$ ($\beta = 1, 2, 3$) authorization against type-II adversaries in Table 2. The security model of type-II adversary, the adversary A_2 submits a target which he/she wants to challenge before the game.

TABLE 1

The security models for the type-I adversary
Experiment EXP^{OW-CCA}_{S,A_1}
$(P_k, S_k) \leftarrow KeyGen(1^k)$
$M \leftarrow A_1^{O_1,O_2,O_3}.(i, P_k)$
$CT^* \leftarrow Enc(P_k, M)$
$M^* \leftarrow A_2^{O_4,O_5,O_6}.(i, P_k)$
If $M = M^*$ then return 1;
Else return 0.

TABLE 2

The security models for the type-II adversary
Experiment EXP^{OW-CCA}_{S,A_2}
$(P_k, S_k) \leftarrow KeyGen(1^k)$
$M \leftarrow A_1^{O_1,O_2,O_3}.(i, P_k)$
$d^* \leftarrow^R \{0, 1\}$
$CT^* \leftarrow Enc(P_k, M_b)$
$d^* \leftarrow A_2^{O_4,O_5,O_6}.(i, P_k)$
If $d^* = d$ then return 1;
Else return 0.

Here,

$O_1(i) = Key_{Gen}(i, SP)$,

$O_2(i, CT_i) = Dec(Sk_i, CT_i)$,

$O_3(i, .) = ET - Auth(Sk_i, *)$, $O_4(i) = O_1(i) or O_4(i) = \perp$, and the condition is $i \neq K$;

$O_5(i, CT_i) = O_2(i, CT_i)$ or $O_5(i, CT_i) = \delta$, the condition is that $CT_i \neq CT^*$, $O_6 = O_3$. For O_6, when $\beta = 1$, then $i \neq K$; When $\beta = 2$, or $\beta = 3$, then $CT_j \neq CT^*$.

4 | THE PROPOSED SCHEME

In this section, we illustrate the model of the developed system for PKEwET method illustrated in Figure 3. This model includes four various components which includes, Key generation center (KGC), server for cloud (CS), data owners (DO), and data user (DU). KGC develops the secret key which is partial for the users based on the user's identity. Therefore the user sends the message and produces the trapdoor by utilizing the authorization and encryption algorithm, individually. Afterward, DO carries data that are being encrypted and trapdoors to server's cloud. In order to search the data which are encrypted in server, the data are sent to the server using trapdoor. By utilizing the server the difference among the CB and CA is found. Finally, if CB = CA, then server produces the result of the search to the user. Therefore, server return 0. In the proposed method, four different kinds of authorizations are introduced for providing the equality test flexibly.

Let four different user be A, B, D, and E. various types of authorizations are given below:

1. Type 1-authorization level of user: in this authorization, every ciphertext of user A can be analyzed with that of every ciphertext of user E and D.
2. Type 2-authorization level of ciphertext: in this authorization, we analyze a certain ciphertext of user A with certain user E's ciphertext.
3. Type 3-authorization level of ciphertext: in this authorization, we analyze the user A specific ciphertext with all used D's ciphertext. This authorization is analysis of Type 1 and Type 2

Authority of attribute (AA): It is main for establishment of system. It gathers public component and access to secret key, and has collaboration with users to produce the secret key of user's data. This is also known as the attribute collection which is a form of database or the directory in which modification, addition and saving of attributes are securely carried out. It is a trusted foundation of data for making some decision.

Server of could (CS): It is responsible for uploading the stored data and helps the user to search the data for specified ciphertext based on the index, then analyze the results correspondingly.

Outsourced server key generation (OSKG): It consists of user's data which is outsourced for secret key according to public key component and transmits it to AA.

Outsourced server decryption (OSD): After gathering the ciphertext from the user this make the use of secret key for converting ciphertext to plaintext and transmit it back to user.

Authorized server cloud (ASC): It is responsible for analyzing the ciphertext equality based on the algorithm, and analyze the corresponding results.

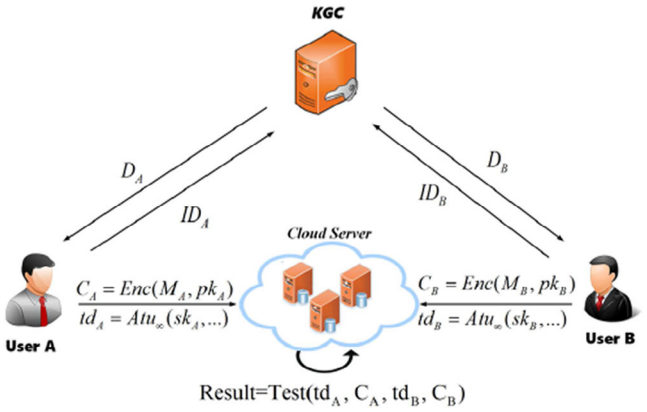


FIGURE 3 Proposed system framework

Data owner (DO): It is used for data encryption, combining with used to produce the ciphertext, and upload to cloud server.

Data user (DU): It is used for producing a keywords token and transmitting it to server cloud. If searching is perfect, then there will be gain in ciphertext, and attain the plaintext. When it is essential for the different ciphertext that contains the similar plaintext, user transmits the ciphertext and trapdoors with the execution of algorithm and gets the user data.

1. **Initial setup:**
   It takes a system parameter μ as input and output the system parameters Sp as follows: Let N be noncommutative near-ring with identities, $N = \ominus(2^{2\mu})$. Let $a, b \in N$ be two noncommuting elements with order $l, k = \theta(2^{2\gamma})$ respectively. We demand the $(a) \cap (b) = \{1\}$ and $H_1 : N \to \{0,1\}^{\mu+2m}$, $H_2 : N^3 \times \{0,1\}^{\mu+2m} \to \{0,1\}^{4m}$ and $H_3, H_4, H_1, H_6 : \{0,1\}^{\mu} \to Z_p^*$, Where m is the length of the elements in Z_p^*. Then the output is $Sp = \{N, a, b, H_1, H_2, H_3, H_4, H_5, H_6\}$.
2. **KeyGen**(Sp)**:**
   It takes system parameters SP as input and outputs: $(P_k, S_k) = ((U = a^{\alpha_1} w b^{\beta_1}, V = a^{\alpha_2} W b^{\beta_2})(a^{\alpha_1}, b^{\beta_1}, a^{\alpha_2}, b^{\alpha_2}))$. Where, $\alpha_1, \alpha_2, \beta_1, \beta_2 \in Z_p^*$ are secure randomly.
3. **Encryption**(M, P_k)**:**
   The function encrypts a message $M \in \{0,1\}^h$ with the public key P_k as follows:
   (a) **Step-1**
   Constructs a straight line $\phi(x)$ and generates two points.
   (b) **Step-2**
   Creates two points, $q_1 = (H_3(M), H_4(M))$ and $q_2 = (H_t(M), H_6(M))$;
   (c) **Step-3**
   Uses two points, q_1 and q_2 to construct a straight line $\phi(Z) = cz + d$;
   chooses $z_1, z_2 \in \{0,1\}^m$ randomly and obtain two points $(z_1, y_1), (z_2, y_2)$ on $\phi(z)$, where $z_1 \neq 0$ or $z_2 \neq 0$.
   (d) **Step-4**
   Choose $l_1, l_2, l_3, l_4 \in Z_p^*$, randomly computes: $C_1 = a^{l_1} w a^{l_2}$, $C_2 = a^{l_3} w a^{l_4}$, $C_3 = M||l_3||l_4 \oplus H_1(a^{l_1} U a^{l_2})$, $C_4 = z_1||z_2||y_1||y_2 \oplus H_2(a^{l_3} V a^{l_4} C_1, C_2, l_3)$ The cipher is, $CT = (C_1, C_2, C_3, C_4)$.
4. **Decrypt**(CT, Sk):
   To decrypt the ciphertext $CT = (C_1, C_2, C_3, C_4)$ with the private key S_k, this algorithm performs the following operations:
   $M||l_3||l_4 = C_3 \oplus H_1(a^{\alpha_1} w C_1, b^{\beta_2}), z_1||z_2||y_1||y_2$
   $= C_4 \oplus H_2(a^{\alpha_2} \omega C_2 b^{\beta_2}, C_1.l_2, C_3)$
   Constructs $\phi(x)$ as in Encrypt($step1$), and verifies as follows: $f(Z_1) = y_1$, $f(y_2) = y_2$ and $C_2 = a^{l_3} w b^{l_4}$. When all the equations output. Otherwise an error message $\perp$ is produced as the output.

4.1 | Four types of authorization algorithm and test algorithm

Assume that there are two users A and B with the ciphertext $CT_i = (C_i, 1, C_i, 2, C_i, 3, C_i, 4)$ (resp. $CJ_i = (Cj.1, C_j.2, C_j.3, Cj.4)$). Let $(l_{i,1}, l_{i,2}, l_{i,3}, l_{i,4})$ $(resp.(l_{j,1}, l_{j,2}, l_{j,3}, l_{j,4})$ be random numbers that are used in $CT_i(resp.CT_j)$.

Type-I Authorization and test

Auth-1

For user A, it takes a part of private key $(a^{\alpha_i,2}, b^{\beta_i,2})$ as input and outputs a trapdoor, $Kr_{1,i} = (a^{\alpha_{i,2}}, b^{\beta_{i,2}})$.

For user B, it takes a part of private key $(a^{\alpha_{j,2}}, b^{\beta_{j,2}})$ as input and outputs a trapdoor $Kr_{2,j} = (a^{\alpha_{i,2}}, b^{\alpha_{j,2}})$.

Test-1 $(CT_i, Kr_1, i, CT_j, Kr_{1,j})$

To test CT_i and CT_j with $Kr_{1,i}$ and $Kr_{1,j}$, the function performs the following operations: $z_{i,1}||z_{i,2}||y_{i,1}||y_{i,2} = C_{i,4} \oplus H_2(a^{\alpha^{i,2}} \omega C_{i,2} b^{\beta}_{i,2}, C_{i,1}, C_{i,2}, C_{i,3})$ $z_{i,1}||z_{j,2}||y_{j,2} = C_{j,4} \oplus H_2(a^{\alpha}_{j,2} \omega C_{j,2} b^{\beta}_{j,2}, C_{j,1}, C_{j,2}, C_{j,1})$ and recovers $\phi(z) = (q, (Z_{i,1}, y_{i,1}).(z_{i,2}, y_{i,2}))$, $\phi_i(z) = (q, (z_{j,1}, y_{j,1}), (z_{j,2}, y_{j,2})$. Finally, it outputs 1 when $\phi_i(z) = \phi_j(z)$ holds. Otherwise, it outputs 0.

Type-2 Authorization and test

Auth-2

For a user A, it takes a part of the private key $(a^{\alpha_{i,2}}, b^{\beta_{i,2}})$ as input and outputs a trapdoor. $Kr_{2,i}, CT_i = H_2(a^{\alpha_{1,2}} \omega C_{i,2} b^{\beta_{i,2}}, C_{i,1}, C_{i,2}, C_{i,3})$.

For user B, it takes a part of the private key $(a^{\alpha_{i,2}}, b^{\beta_{j,2}})$ as input and outputs a trapdoor $Kr_{2,j}, CT_j = H_2(a^{\alpha_{j,2}} \omega C_{j,2} b^{\beta_{j,2}}, C_{j,1}, C_{j,2}, C_{j,3})$.

Test-2 $(CT_i, Kr_{2,i}, CT_i, CT_j, Kr_{2,j}, CT)$

To test CT_i and CT_j with $Kr_{2,i}$, CT_i and CJ_i with Kr_{2,i,CT_i} and Kr_{2,i,CT_j} the function performs the following operations: $z_{i,1}||x_{i,2}||y_{i,1}||y_{i,2} = C_{i,4} \oplus Kr_{2,i,CT_i}, z_{j,1}||z_{j,2}||z_{j,2} = C_{j,4} \oplus Kr_{2,j}, CT$; and recovers $\phi_i(z) = (q_1(z_i.1, y_i.1), (z_i, 2, y_i, 2))$ Finally, it outputs 1 when $\phi_i(z) = \phi_j(z)$

Type-3 Authorization and test

Let z be a bit-string then $[z]_a^b$ defines a sub string of z, from a to b.

Auth-3

For user A, it takes the straight line $\phi_i(z) = c_i z_i + d_i$, $z_{i,1}, z_{1,2}, l_{i.3}, l_{i,4}$, which is used in the encryption algorithms as a part of the private key $(a^{\alpha_{i,2}}, b^{\beta_{i,2}})$, then it outputs a trapdoor, $Kr_{3,i,CT_{i,j},CT} = (Q_i, X_i) = ([H_2(a^{\alpha_{i,2}}\omega C_{i,2} b^{\beta_{i,2}}, C_{i,1}, C_{i,2}, C_{i,3}]_{2l}^{4i-1}$, $a^{\alpha_i z_{i,1}} q_i b^{\alpha_i z_{i,1}}$, Where $q_i = a^{l_{i,3}} \omega C_{j,2} b^{l_{i,4}}$).

For user B, it takes the straight line $\phi_i(z) = C_j z_j + b_j, z_j, 1, z_j, 2, l_j, 3, l_j, 4$. While it is used in encryption algorithm and a part of the private key $(a^{\alpha_i,2}, b^{\beta_i,2})$ then it outputs a trapdoor. $Kr_{3,j,CT_j,i,CT_i} = (Q_j, X_j) = ([H_2(a^{\alpha_{i,2}}\omega C_{j,2} b^{\beta_{j,2}}, C_{j,1}, C_{j,2}, C_{j,i}]_{2l}^{4l-1}$, $a^{\alpha_j z_j,1} w q_j b^{\beta_i z_i,2}$. Where, $q_j = a^{l_j,3} \omega C_{i,2} b^{l_{j,4}}$.

Test-3 $(CT_j, Kr_{3,i,CT_i j,CT_j,CT_j}, CT_j, Kr_{3,j,CJ_i,i,CT_i})$

This algorithm works as follows: $y_{i,1}||y_{i,2} = [(C_{i,4})_{2l}^{4l-1} \oplus Q_i$, $y_{j,1}||y_{j,2} = [C_{j,4}]_{2l}^{4l-1} \oplus Q_j$. Then, it computes $\psi_i = a^{-y_{i,1}} \omega X_i b^{-y_{i,2}}$, $\psi_j = a^{-y_{i,1}} \omega X_j b^{-y_{j,2}}$. Finally, it outputs 1 when $\psi_i = \psi_j$ holds. Otherwise, it outputs 0.

Type-4 Authorization and test

Auth-4

For user A, this function takes a part of the private key $(a^{\alpha_{i,2}}, b^{\alpha_{i,2}})$ as input and outputs a trapdoor $Kr_{4,i,CT_i} = Kr_{2,i,CT_i} = H_2(a^{\alpha_{i,2}} \omega C_{i,2} \beta^{\beta_{i,2}}, C_{i,1}, C_{i,2}, C_{i,3})$.

For user B, it takes a part of the private key $(a^{\alpha_j,2}, \beta^{\beta_j,2})$ as input and outputs a trapdoor $Kr_{4,j} = Kr_{1,j} = (a^{\alpha_{j,2}}, b^{\beta_{i,2}})$.

Test-4 $(CT_j, Kr_{4,i,CT_i}, CT_j, Kr_{4,j})$

To test CT_i and CT_j, with Kr_{2,i,CT_i} and Kr_{2,i,CT_j}, it works as follows: $z_{i,1}||z_{i,2}||y_{i,1}||y_{i,2} = C_{i,4} \oplus Kr_{2,i,CT_i}\, z_{j,1}||z_{i,2}||y_{j,1}||y_{j,2} = C_{j,4} \oplus H_2(a^{\alpha_{j,2}} \omega C_{j,2} b^{\beta_{j,2}}, C_{j,1}, C_{j,2}, C_{j,3})$ and resolves: $\phi_i(z) = (q, (z_{i,1}, y_{i,1}), (z_{i,2}, y_{i,2}), \phi_j(z) = (q, (z_{j,1}, y_{j,1}), (z_{j,2}, y_{j,2})$ Finally, it outputs 1 when $\psi_i = \psi_i$ holds. Otherwise, it outputs 0.

5 | SECURITY ANALYSIS

5.1 | Theorem

The proposed cryptosystem is $OW - CCA$ secure in the case that the $CDH_{a,b}^N$, problem is hard in the random oracle model for the type-I adversary.

Proof. Suppose that there is a type-1 adversary A_1 could break the $OW - CAA$ security of the choul cryptosystem in polynomial time. Now we construct an algorithm δb solve the $CDH_{a,b}^N$ problem in N with nonnegligibility. Given a 4-tuple $(a, b, a^{\alpha 1} w b^{\beta 1}, a^{\alpha 2} w b^{\beta 2}) \in N$, the objective of the algorithm δ is to compute $a^{\alpha 3} w b^{\beta 3} = a^{\alpha_1 + \alpha_2} w \beta^{\beta_1 + \beta_2}$. The game between δ and A_1 runs as follows, during the interaction, δ will store its response to all queries and hold the H_1 with list. ∎

Init: The adversary A_1 submits a target μ that he or she wants to change before the game.

Setup: The simulator generates the system parameter (S_p) with a security parameter δ as in the algorithm setup. Then it generates n-pairs of public/ private key as follows:

For each $1 \le j \le n (j \ne K)$. It choose a random $\alpha_i, 1, \alpha_i, 2, \beta_{i,1}, \beta_{i,2} \in Z_P^*$ and computes $U_j = a^{\alpha_{i,1}} w b^{\beta_{i,1}}$, $V_i = a^{\alpha_{i,2}} \omega b^{\beta_i,2}$. The public key is (U_i, V_j), and the private key is $(a^{\alpha_{i,1}}, b^{\beta_{i,1}}, a^{\alpha_{i,2}}, b^{\beta_{i,2}}, w)$. If $i = K$, Let $U_k = a^{\alpha_1} w b^{\beta_1}$, $V_K = a^{\alpha_k,2} w b^{\beta k,2}$. While δ can only be the private key. $V_k = a^{\alpha k,2} w b^{\beta k,2}$. Note that δ has no information about the private key of U_K. Then, the simulation provides public keys to A_1.

Phase-1

After receiving the public key, A_1 can perform the decryption key queries, decryption queries, authorization queries and random oracle H_1 queries. The game between A_1 and δ runs as follows:

$O_{H\,1}$ **-queries:**

- A_1 can perform the O_{H_1}−queries adaptively. To respond to the queries, δ holds a list of tuple $H_1 - (\gamma_i, \sigma_i)$ and respond as follows:

- When γ_i is already in H_1 in the tuple (γ_i, σ_i) then δ outputs $H_1(\gamma_i) = \sigma_i$ as the answer.
- Otherwise, δ choose $\sigma_i \in \{0,1\}^{\mu+2m}$ randomly, stores a new tuple (γ_i, σ_i) in to the H_1 lists outputs $H_1(\gamma_i) = \sigma_i$ as the answer.

Decryption key queries: A_1 can perform the decryption key queries adaptively and δ responds A_1 with Sk_i, which is generated during $(i \neq K)$.

Decryption queries: Let $CT_i = (C_{i,1}, C_{i,2}, C_{i,3}, C_{i,4})$. If $i \neq K$, δ calls the decryption key queries for Sk_i. Then, it runs decryption algorithm and output the answer to A_1. Otherwise, δ answers as follows:

If each tuple (γ_i, σ_i) is in the H_1-list, δ_1 calculates:

Step-1

$M_i||l_{i,3}||l_{i,4} = C_{i,3} \oplus H_1(\gamma_i)$

$z_{i,1}||z_{i,2}||y_{i,1}||y_{i,2} = C_{i,4} \oplus H_2(a^{\alpha_{i,2}}\omega C_{i,2} f^{\beta_{i,2}}, C_{i,1}, C_{i,2}, C_{i,3})$

Step-2

It generates $q_{i,1}, q_{i,2}$ as the encrypt algorithm by using M_1.

Step-3

It constructs a straight line $\phi_i(z)$, by using the two points $q_{i,1}, q_{i,2}$ generated during step 2.

Step-4

δ returns M_i, if $\phi_i(z_{i,1}) = y_{i,1}$ and $\phi_i(x_{i,2}) = y_{i,2}$ and $C_{i,2} = a^{l_{i,3}}\omega b^{l_{i,4}}$ holds. Otherwise, it outputs $\perp$ to A_1.

Authorization querying: For the type-β $(\beta = 1, 2, 3)$ authorization, it runs as follows:

Step-1

When $\beta = 1$,with input i, δ runs the auth-1 algorithm and answers A_1 with $Kr_{1,i} = (a^{\alpha_{i,2}}, b^{\beta_{i,2}})$.

Step-2 When $\beta = 2$, with input (i, CT_i), δ runs the auth-2 algorithm and answer A_1 with $Kr_{2,i,CT_i} = H_2(a^{\alpha_{i,2}}\omega C_{i,2} b^{\beta_{i,2}}, C_{i,1}, C_{i,2}, C_{i,3})$.

Step-3 Where $\beta = 3$, with input (i, CT_i, j, CT) δ runs auth-3 algorithms and answer, A_1 with $Kr_3, i, CT_i, j, CT_j = (\phi_i, X_i) = ([H_2(a^{\alpha_{i,2}}\omega C_{i,2} b^{\beta_{i,2}}, C_{i,1}, C_{i,2}, C_{i,3})]_{2l}^{4l-1}, a^{\alpha_i z_{i,1}} q_i b^{\beta_i z_{i,2}})$, Where $q_i = a^{l_{i,3}}\omega C_{j,2} b^{l_{i,4}}$.

Challenge: δ select a message M_K randomly with a challenge $l_{K,i}, l_{K,2} \in Z_p^*$. Then it outputs $CT_K = (C_{K,1}, C_{K,2}, C_{K,3}, C_{K,4})$ as follows: $C_{K,1} = a^{\alpha_2}\omega b^{\beta_2}$, $C_{K,2} = a^{l_{K,1}}\omega b^{l_{K,2}}$, $C_{K,3} = M||l_\mu, 1||l_\mu, 2 \oplus H_1(a^{\alpha_3} w b^{\beta_3})$, $C_{K,3} = M||l_K, 1||l_K, 2 \oplus H_1(a^{\alpha 3} w b^{\beta_3})$, $C_K, 4 = z_1||z_2||y_1||y_2 \oplus H_2(a^{l_{K,1}} V_K b^{l_{K,2}}, C_{K,1}, C_{K,2}, C_{K,3})$.

Finally, it gives CT_K to A_1 at the challenge ciphertext.

Phase-2

A_1 continues to perform query in phase 1. The restrictions are as follows:

- In the decryption key query $i \neq k$.
- In the decryption query, CT_k is not allowed.

Guess: Finally assumes a guess M^*. If $M^* = M_k$ holds and δ outputs $a^{\alpha 3}\omega b^{\beta 3} = a^{\alpha_1+\beta_1}\omega b^{\alpha_2+\beta_2}$ as the answer.

5.2 | Theorem

The proposed cryptosystem is IND-CCA secure in the case when the $DDH_{a,b}^N$, problem is hard in the random oracle model for type-II adversary.

Proof. The proposed cryptosystem is IND-CCA secure in the case when the $DDH_{a,b}^N$, problem is hard in the random oracle model for type-II adversary. proof. Suppose that there is a type-II adversary A_2 that can break the IND-CCA security of the above cryptosystem in polynomial-time. Now, we construct an algorithm δ to solve the $DDH_{a,b}^N$, problem in N with nonnegligible probability. Given a 5-tuple $(a, b, a^{\alpha_1}\omega b^{\beta_1}, a^{\alpha_3}\omega b^{\beta_3}) \in N$ the object of algorithm δ is to test whether $a^{\alpha_3}\omega b^{\beta_3} = a^{\alpha_1+\alpha_2}\omega b^{\beta_1+\beta_2}$ holds. The game between δ and A_2 runs as follows. During the interaction, δ will store its responses to all queries and hold the H_1 watch list. ∎

Init: The adversary A_2 submits a target K that he/she wants to challenge before the game.

Setup: The simulator generates the system parameters sp with a security parameter β as in the algorithm setup. Then, it generates n-pairs of public/private keys as follows:

For each $1 \le i \le n (i \ne K)$, it chooses a random $\alpha_{i,1}, \alpha_{i,2}, \beta_{i,1}, \beta_{i,2} \in Z_p^*$ and computes $U_K = a_{i,1}^{\alpha} \omega b_{i,1}^{\beta}, V_K = a_{i,2}^{\alpha} \omega b_{i,2}^{\beta}$. The public key is $(U_K, V_K)$and the private key is $(a_{i,1}^{\alpha}, b_{i,1}^{\beta}, a_{i,2}^{\alpha}, b_{i,2}^{\beta}, \omega)$. If $i \ne K$, let $U_K = a_1^{\alpha} \omega b_1^{\beta}, V_K = a_{k,2}^{\alpha} \omega b_{k,2}^{\beta}$ while δ only has the private key $V_K = a_{k,2}^{\alpha} \omega b_{k,2}^{\beta}$. Notice that δ has no information about the private key U_K. Then, it gives the public keys to A_2.

Phase 1

After receiving the public key, A_2 can perform decryption key queries, decryption queries, authorization queries, and random oracle H_1 queries. The game between A_2 and δ runs as follows:

O_{H_1}-queries: A_2 can perform the O_{H_1}-queries adaptively. To respond to the queries, δ holds a list of tuples-$H_1 - (\gamma_i, \sigma_i)$ and responds as follows:

- If there is iAE' σ_i already in H_1 in the tuple $H_1(\gamma_i, \sigma_i)$, then, outputs $H_1(\gamma_i) = \sigma_i$ as the answer.
- Otherwise, δ chooses $\tau_i \in \{0,1\}^{\mu+2m}$ randomly, stores a new tuple (σ_i, τ_i) into H_1-list and outputs $H_1(\gamma_i) = \sigma_i)$ as the answer.

Decryption key queries: A_2 can perform decryption key queries adaptively and δ responds to A_2 with S_{K_i} which is generated in the Setup $i \ne K$.

Decryption queries: Let $CT_i = (C_{i,1}, C_{i,2}, C_{i,3}, C_{i,4})$.

If $i \ne K$, δ calls the decryption key queries for S_{K_i}. Then it runs decryption algorithm and outputs the answer to A_2.

Otherwise, δ answers as follows: If each tuple (σ_i, τ_i) is in H_1-list, δ calculates:

1. $M_i||l_{i,3}||l_{i,4} = C_{i,3} \oplus H_1(\sigma_i) z_{i,1}||z_{i,2}||y_{i,1}||y_{i,2} = C_{i,3} \oplus H_2(a^{\alpha_{i,2}} \omega b^{\beta_{i,2}}, C_{i,1}, C_{i,2}, C_{i,3})$
2. It generates $q_{i,1}, q_{i,2}$ as in the Encrypt algorithm by using M_i;
3. It constructs a straight line $\phi_i(z)$ by using the two points $q_{i,1}, q_{i,2}$ generated in the step 2;
4. δ returns M_i, if $\phi_i(z_{i,1}) = y_{i,1}, \phi_i(z_{i,2}) = y_{i,2}$ $C_{i,2} = a^{l_{i,3}} \omega b^{l_{i,3}}$ all hold. Otherwise, it outputs $\perp$ to A_2.

Authorization queries: For type-$\beta (\beta = 1, 2, 3)$ authorization, it runs as follows:

1. When $\beta = 1$,, with input i, δ runs the Auth-1 algorithm and answers A_2 with $Kr_{1,i} = (a^{\alpha_{i,2}}, a^{\alpha_{i,2}})$;
2. When $\beta = 2$, with input (i, CT_i), δ runs the Auth-2 algorithm and answers A_2 with $Kr_{2,i,CT_i} = H_2(a^{\alpha_{i,2}} \omega C_{i,2} b^{\beta_{1,2}} C_{i,1}, C_{i,2}, C_{i,3})$;
3. When $\beta = 3$,, with input, (i, CT_i, CT_j), δ with input runs the Auth-3 algorithm and answers A_2 with $Kr_{3,i,CT_i,i,CT_j} = (Q_i, X_i) = ([H_2(a^{\alpha_{i,2}} \omega C_{i,2} b^{\beta_{1,2}} C_{i,1}, C_{i,2}, C_{i,3})]_{2l}^{4l-1}, a^{\alpha_i, z_{i,1}} \omega q_i b_{a_i}^{\beta}, z_{i,2})$, where $q_i = a^{l_{i,3}} C_{j,2} b^{l_{i,4}}$.

Challenge: A_2 takes two messages M_0, M_1 randomly to δ. Then, δ selects a random bit $b \in \{0,1\}$ and chooses $l_{i,1}, l_{i,2} \in Z_p^*$. Then, it outputs $CT_K = C_{i,1}, C_{i,2}, C_{i,3}, C_{i,3}$ as follows:

$C_{K,1} = a^{\alpha_2} \omega b^{\beta_2}$,

$C_{K,2} = a_{l_{K,1}}^{\alpha} \omega b_{l_{K,2}}^{\beta}, C_{K,3} = M_b||l_{K,1}||l_{K,2} \oplus H_1(a^{\alpha_3} \omega b^{\beta_3})$,

$C_{K,4} = z_1||z_2||y_1||y_2 \oplus$

$H_2(a^{l_{K,1}} \omega V_K b^{l_{K,2}}, C_{K,1}, C_{K,2}, C_{K,3})$

Finally, it gives CT_K to A_2 as the challenge ciphertext.

Phase 2 A_2 continues performing queries as in Phase 1. The restrictions are as follows:

- In the decryption key queries, $i \ne K$;
- In the decryption queries, CT_K is not allowed.
- For type-$\beta (\beta = 1, 2, 3)$ authorization queries:

1. When $\beta = 1, i \ne K$;
2. When $\beta = 2, (K, CT_K)$ is not allowed;
3. When $\beta = 3, (K, CT_K, .)$ is not allowed;

Guess. Finally, A_2 submits a guess d^*. If $d^* = d$ holds, δ outputs 1 meaning $a^{\alpha_3} \omega b^{\beta_3} = a^{\alpha_1 + \alpha_2} \omega b^{\beta_1 + \beta_2}$; otherwise, it outputs 0.

6 | PERFORMANCE ANALYSIS

Here, we analyze the efficiency of our scheme. In Figure 4, we provide comparisons to other PKEwET schemes without flexible authorization. From the second column to the fifth column, we show complexity comparisons of the encryption algorithm **(Enc)**, decryption algorithm **(Dec)**, authorization algorithm **(Auth)**, and test algorithm **(Test)**. The last column shows the ability to resist quantum attacks **(RQA)**. Figure 5 shows comparisons to the schemes in Reference 19 with flexible authorization. The second to the fifth column show complexity comparisons of four types of authorization algorithms, while the sixth to the ninth column show.

- *i*-means modular inversion
- *m*-means modular multiplication
- *p*-means pairing evaluation

The proposed algorithm is tested with different existing algorithms on the basis of encryption, decryption, authorization, and test with respect to time cost, where the proposed algorithm has higher authorization time cost when compared with existing approaches. Some others systems as shown in graph has null authorization. The testing time cost for the proposed method has lower time cost. Comparison of time cost is shown in Figure 6.

Figure 7 shows comparisons to the scheme in H. Qu et al. The second to the fifth row shows the complexity comparisons of Enc, Dec, Auth, and Test. The last line shows the ability to resist RQA. From the comparisons in Figure 4, we can

PKEwET	Enc	Dec	Auth	Test	RQA
Ma (2016)	3e	3e	0	2p	NO
Qu et al. (2018)	4e+1m	2e	3e	4p+2i	NO
Wu et al. (2018)	5e	2e	0	2e+2m+2i	NO
Ma et al. (2015)-1	4e+1m	2e	3e	4p+2i	NO
Ma et al. (2015)-2	6e+2p	2e+2p+1i	0	4p+2i	NO
Ma et al. (2015)-3	6e+1m	5e+1m	0	4p+2i	NO
Ma et al. (2015)-4	6e+2p	2e	Type-4	Type-4	NO
Ours	4e+1i+6m	2e+1i+5m	Type-4	Type-4	Yes

FIGURE 4 The comparison of computational complexity (time)

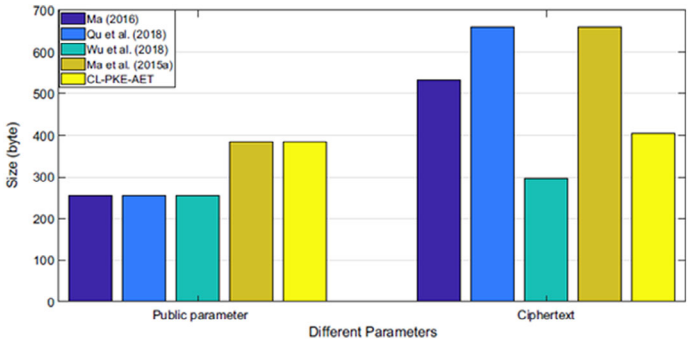


FIGURE 5 Comparison of the time-cost for encryption, decryption, authorization, and test algorithms

PKEwET	Authentication				Test			
	Type-1	Type-2	Type-3	Type-4	Type-1	Type-2	Type-3	Type-4
S. Ma et al	0	2e	2e+2p	2m	2e+2p+m	2e+2p+m	2e+2p+2i+m	2e+2p+m
Our Scheme	0	4m	2e+2i+8m	2m	2i+4m	2i	4e+4i+4m	2i+2m

FIGURE 6 The comparison of computational complexity (time)

FIGURE 7 The comparison of the scheme in H. Qu et al

PKEwET	Enc	Dec	Auth	Test	RQA
H. Qu et al	4e	2e	X	X	Yes
Ours Scheme	4e+i+6m	2e+i+5m	Type-3	Type-3	Yes

FIGURE 8 Search Time

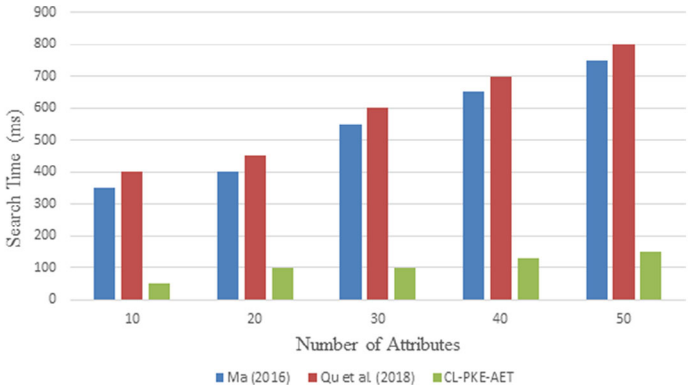


FIGURE 9 Decryption time

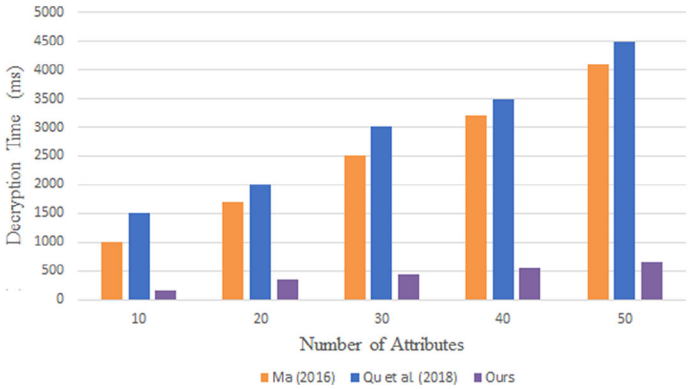
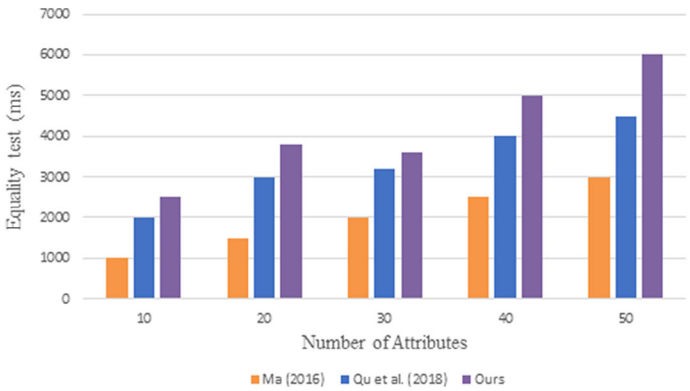


FIGURE 10 Equality test



see that there is no authorization in because these algorithms do not use different types of authorizations according to different situations. In Ma added four types of authorization policies. Figure 6 shows that our scheme is more efficient in Enc and Dec.

Security enhanced using the method is evaluated using different time metrics. Search time is one such metric which uses the attributes numbers based on the performance of the systems in providing security. Figure 8 describes the search time comparison made with different existing algorithms. Where our proposed security based method out performs other by minimizing the search time with increase in number of attributes.

Figure 9 shows the illustration of various existing algorithm with proposed system for decryption time. With increase in number of attributes, the time of decryption decreases. The proposed methodology has efficient decryption time

where lesser the time consumed for decryption is higher the performance of the proposed system. Public key encryption is used for securely transmitting the user's data to the recipient using near-ring. This transmission is analyzed using equality testing which is shown in Figure 10, where our proposed method makes the security enhanced using near-ring which is compared with existing algorithms. Our approach for security enhancement increases with total number of attributes.

7 | CONCLUSION

IoT is emerging technology in various aspects of today's world, with increase in security concern, transmission of data should be made more secure to avoid unauthorized access to confidential information. This article describes authorized equivalence scheme by utilizing significant public key cryptography. Authorized mechanism is our suggested scheme which is efficient in flexibility such as capability in testing by authorized users enabled to access the cloud server with various authorization schemes. Moreover, procedure of transmitting data takes places using oracle random model, based on the assumption of DLPDCP the proposed security model has proved its efficiency. Thus, providing security in IIoT based model remains as a greater challenge. In this proposed model, PKEwET based near-ring model is developed for Industrial IoT system. Search time of the proposed scheme is 150 milliseconds for which the number of attributes is 50 and when comparing to the decryption time of the proposed model which is lower when compared to other existing scheme for 50 attributes. Our proposed privacy enhancement is proved to be efficient against random Oracle model and quantum algorithm.

ACKNOWLEDGMENT

This work was partially supported by the National Natural Science Foundation of China (Grant No. 61872084); Guangdong-Hong Kong-Macao Intelligent Micro-Nano Optoelectronic Technology Joint Laboratory (Project No. 2020B1212030010). Open access funding enabled and organized by Projekt DEAL.

DATA AVAILABILITY STATEMENT

Data sharing not applicable as no new data were created or analyzed in this study.

ORCID

Ganesh Gopal Deverajan https://orcid.org/0000-0003-0036-7841
Ching-Hsien Hsu https://orcid.org/0000-0002-2440-2771

REFERENCES

1. Abdalla M, Bellare M, Catalano D, et al. Searchable encryption revisited: consistency properties, relation to anonymous IBE, and extensions. Crypto 3621; 2005:205-222; Springer, New York, NY.
2. Al-Riyami SS, Paterson KG. Certificateless public key cryptography. Asiacrypt 2894:452–473 (Springer) Anagnostopoulos I, Zeadally S, Exposito E (2016) Handling big data: research challenges and future directions. *J Supercomput*. 2003;72(4):1494-1516.
3. Atzori L, Iera A, Morabito G. The internet of things: a survey. *Comput Netw*. 2010;54(15):2787-2805.
4. Baek J, Safavi-Naini R, Susilo W. Public key encryption with keyword search revisited. *Comput Sci Appl ICCSA*. 2008;2008:1249-1259.
5. Baza, M, Lasla, N, Mahmoud, M, Srivastava, G, & Abdallah, M. (2019). B-ride: ride sharing with privacy-preservation, trust and fair payment atop public blockchain. *IEEE Trans Netw Sci Eng*. http://dx.doi.org/10.1109/TNSE.2019.2959230.
6. Perera C, Zaslavsky A, Christen P, Georgakopoulos D. Sensing as a service model for smart cities supported by Internet of Things. *Trans Emerg Telecommun Technol*. 2014;25(1):81-93.
7. Sharma G, Kuchta V, Anand Sahu R, et al. A twofold group key agreement protocol for NoC-based MPSoCs. *Trans Emerg Telecommun Technol*. 2019;30(6):e3633.
8. Gupta D, Khanna ASKL, Shankar K, Furtado V, Rodrigues JJ. Efficient artificial fish swarm based clustering approach on mobility aware energy-efficient for MANET. *Trans Emerg Telecommun Technol*. 2019;30(9):e3524.
9. Khalili S, Simeone O. Inter-layer per-mobile optimization of cloud mobile computing: a message-passing approach. *Trans Emerg Telecommun Technol*. 2016;27(6):814-827.
10. Boneh D, Di Crescenzo G, Ostrovsky R, Persiano G. Public key encryption with keyword search. Paper presented at: Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques; 2004:506-522; Springer, New York, NY.
11. Dwivedi AD, Malina L, Dzurenda P, Srivastava G. Optimized blockchain model for internet of things based healthcare applications. Paper of the: Proceedings of the 2019 42nd International Conference on Telecommunications and Signal Processing (TSP), Budapest, Hungary on July 1-3, 2019:135-139; IEEE.

12. Elhabob R, Zhao Y, Sella I, Xiong H. Efficient certificateless public key cryptography with equality test for internet of vehicles. *IEEE Access*. 2019;7:68957-68969.
13. Esposito C, Castiglione A, Martini B, Choo KKR. Cloud manufacturing: security, privacy, and forensic concerns. *IEEE Cloud Comput*. 2016;3(4):16-22.
14. Gope P, Das AK, Kumar N, Cheng Y. Lightweight and physically secure anonymous mutual authentication protocol for realtime data access in industrial wireless sensor networks. *IEEE Trans Ind Inform*. 2019;15(9):4957-4968.
15. Hassan A, Eltayieb N, Elhabob R, Li F. An efficient certificateless user authentication and key exchange protocol for client-server environment. *J Ambient Intell Humaniz Comput*. 2018;9(6):1713-1727.
16. Huang K, Tso R, Chen YC, Rahman SMM, Almogren A, Alamri A. Pke-aet: public key encryption with authorized equality test. *Comput J*. 2015;58(10):2686-2697.
17. Kim S, Lee I. Iot device security based on proxy re-encryption. *J Ambient Intell Humaniz Comput*. 2018;9(4):1267-1273.
18. Lee Hyung Tae, Ling San, Seo Jae Hong, Wang Huaxiong. Semi-generic construction of public key encryption and identity-based encryption with equality test. *Information Sciences*. 2016;373:419–440. http://dx.doi.org/10.1016/j.ins.2016.09.013.
19. Li Depeng, Aung Zeyar, Sampalli Srinivas, Williams John, Sanchez Abel. Privacy Preservation Scheme for Multicast Communications in Smart Buildings of the Smart Grid. *Smart Grid and Renewable Energy*. 2013;04(04):313–324. http://dx.doi.org/10.4236/sgre.2013.44038.
20. Ma S. Identity-based encryption with outsourced equality test in cloud computing. *Inf Sci*. 2016;328:389-402.
21. Ma S, Huang Q, Zhang M, Yang B. Efficient public key encryption with equality test supporting flexible authorization. *IEEE Trans Inf Forens Secur*. 2015;10(3):458-470.
22. Ma S, Zhang M, Huang Q, Yang B. Public key encryption with delegated equality test in a multi-user setting. *Comput J*. 2015;58(4):986-1002.
23. Molano JIR, Lovelle JMC, Montenegro CE, Granados JJR, Crespo RG. Metamodel for integration of internet of things, social networks, the cloud and industry 4.0. *J Ambient Intell Humaniz Comput*. 2018;2018:1-15.
24. Noroozi M, Eslami Z. Public-key encryption with keyword search: a generic construction secure against online and offline keyword guessing attacks. *J Ambient Intell Humaniz Comput*. 2019;2019:1-12.
25. Premkamal PK, Pasupuleti SK, Alphonse P. A new verifiable outsourced ciphertext-policy attribute based encryption for big data privacy and access control in cloud. *J Ambient Intell Humaniz Comput*. 2018;2018:1-15.
26. Sakhnini J, Karimipour H, Dehghantanha A, Parizi RM, Srivastava G. Security aspects of Internet of Things aided smart grids: A bibliometric survey. *Internet of Things*. 2019;100111. http://dx.doi.org/10.1016/j.iot.2019.100111.
27. Singh S, Sharma PK, Moon SY, Park JH. Advanced lightweight encryption algorithms for iot devices: survey, challenges and solutions. *J Ambient Intell Humaniz Comput*. 2017;2017:1-18.
28. Tang Q. Towards public key encryption scheme supporting equality test with fine-grained authorization. Paper presented at: Proceedings of the Australasian Conference on Information Security and Privacy, Springer, New York, NY; 2011:389-406. http://dx.doi.org/10.1007/978-3-642-22497-3_25.
29. Zhou Y, Li N, Tian Y, An D, Wang L. Public key encryption with keyword search in cloud: a survey. *Entropy*. 2020;22(4):421.
30. Malina L, Srivastava G, Dzurenda P, Hajny J, Fujdiak R. A secure publish/subscribe protocol for internet of things. Paper presented at: Proceedings of the 14th International Conference on Availability, Reliability and Security; 2019:1-10; Springer, New York, NY.
31. Tang Q. Public key encryption schemes supporting equality test with authorisation of different granularity. *Int J Appl Cryptogr*. 2012;2(4):304-321.
32. Tang Q. Public key encryption supporting plaintext equality test and user-specified authorization. *Secur Commun Netw*. 2012;5(12):1351-1362.
33. Thirumalai C, Mohan S, Srivastava G. An efficient public key secure scheme for cloud and IoT security. *Comput Commun*. 2020;150:634-643.
34. Truong HL, Dustdar S. Principles for engineering IoT cloud systems. *IEEE Cloud Comput*. 2015;2(2):68-76.
35. Xu X, Han M, Nagarajan SM, Anandhan P. Industrial Internet of Things for smart manufacturing applications using hierarchical trustful resource assignment. *Comput Commun*. 2020;160:423-430.

How to cite this article: Deverajan GG, Muthukumaran V, Hsu C-H, Karuppiah M, Chung Y-C, Chen Y-H. Public key encryption with equality test for Industrial Internet of Things system in cloud computing. *Trans Emerging Tel Tech*. 2021;e4202. https://doi.org/10.1002/ett.4202