Journal of Theoretical and Applied Information Technology
30th November 2021. Vol.99. No 22

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

IOT BASED RUZICKA INDEX TWO-STEP AUTHENTICATIVE MATYAS–MEYER MESSAGE DIGEST FOR SECURED CLOUD DATA STORAGE WITH INDUSTRIAL INTERNET OF THINGS

FLINDON W.G [1], DR. DIVYA S.V [2]

[1]Research Scholar, Department Of Computer Application,
Noorul Islam Centre For Higher Education, Thuckalay, Kumaracoil, Tamil Nadu 629180, India.
[2]Assistant Professor, Department Of Information Technology,
Noorul Islam Centre For Higher Education, Thuckalay, Kumaracoil, Tamil Nadu 629180.

E-mail: [1] mrflindon@gmail.com, [2] divyasadasivam@gmail.com

ABSTRACT

Cloud computing is a kind of computing technology based on shared resources than the local servers to handle the different applications. Cloud computing is the process of storing data on remote servers and accessing them through the internet. Cloud provides a large amount of virtual storage to users. Security is one of the essential barriers for protecting the stored data from unauthorized user access. However, existing methods failed to improve authentication accuracy and data confidentiality. In order to improve the security level during the data storage in cloud computing, a novel technique called IoT based Ruzicka Index Two-Step Authenticative Matyas–Meyer Message-Digest Cryptography (IoT-RITSAMMMDC) is introduced in industrial applications. The main aim of IoT-RITSAMMMDC is to store the industrial data with a higher confidentiality rate, integrity, and lesser space complexity. The IoT-RITSAMMMDC consists of three major processes namely registration, authentication, and secure data storage. In the registration phase, the user's details like name, age, date of birth, etc are to be registered to the cloud server. After registering the user details, the cloud server generates an identity (ID) and password for every registered cloud user. Cloud server stores user ID and password for performing the two-step authentication using Ruzicka similarity index. Whenever the cloud user needs to store the data, the cloud user sends the ID to the server for performing the authentication process. The cloud server verifies whether the user is authorized or not by matching the cloud user ID and password. After authentication, the cloud user is allowed to store the data in a cloud server. In the IoT-RITSAMMMDC technique, the cloud server uses Merkle–Damgård Matyas–Meyer Message-Digest Cryptographic hash for storing the cloud user data in a secured manner. This in turn helps to improve the data confidentiality and reduce the space complexity. Experimental evaluation is carried out on the factors such as authentication accuracy, data confidentiality, data integrity rate, and space complexity with respect to a number of cloud user requests and data. The observed results reveal that our proposal IoT-RITSAMMMDC technique offers an efficient solution in terms of achieving higher authentication accuracy, data confidentiality, data integrity rate, and lesser space complexity.

Keywords: *Cloud Computing*, *Industrial Iot*, *Secured Data Storage*, *Registration*, *Two-Step User Authentication, Ruzicka Similarity Index, Merkle–Damgård Matyas–Meyer Message Digest*

1. INTRODUCTION

Cloud computing provides a large storage area where resources are available everywhere at any time. Cloud computing is a type of computing based on shared resources than the local servers to handle various IoT applications such as healthcare, industry, agriculture, smart homes, and so on. Recently, IoT technology has become a preferred tool in the industrial field for preventing sensitive information like product quality defects, especially in production automation and smart factories. Security is the major concern for continuing production growth and to preserve the data from unauthorized user access. In order to improve the security level during the data storage in cloud computing, cryptosystem techniques have been developed.

A Secure Auditable Cloud Storage (SecACS) mechanism was introduced in [1] for secure cloud

storage and supporting data dynamics. SecACS focus on verifying the data integrity with less computation time but the higher confidentiality rate was not achieved. A multi-security-level cloud storage system was introduced in [2] for protecting the users' private data using improved proxy re-encryption. But, it failed to consider the authentication for enhancing the confidentiality of data storage in the cloud.

A dynamically updatable privacy-preserving authentication approach was introduced in [3] for general directed graphs to intuitively represent the relationships. The authentication scheme was designed depending on the cryptographic accumulator and standard digital signature scheme. But, the scheme was probabilistic and inefficient for IIoT systems.

A Lightweight security mechanism was designed [4] for communications in Industrial IoT depending on the hash and XOR processes for M2M communications in the Industrial IoT environment. But, the authentication mechanism failed to provide lightweight mutual authentication between the sensors in the Industrial IoT environment. A hybrid framework was introduced in [5] with the software and hardware integration policy for the industrial platform. But, authentication accuracy and data confidentiality was not improved by the presented approach.

A novel security model of a cloud storage auditing protocol was designed in [6] to enables the integrity of the data. But the model failed to minimize the storage overhead for achieving a better security level. An Elliptic Curve Cryptography was introduced in [7] for generating highly secured keys to enhance the data integrity and secure storage. However, the authentication was not performed. A framework consists of dual encryption and data fragmentation was developed in [8] that predict the secure distribution of information. But the higher data confidentiality rate was not achieved.

An enhanced group-based cryptography method was introduced in [9] for secure data storage with minimum computation time. But it failed to perform the authentication for increasing the data confidentiality. An adaptively secure system was introduced in [10] for certificate-based broadcast encryption to enhance security. However, the system failed to construct more efficient schemes for reducing the space complexity.

1.1 Research Contributions

In this paper, the above existing drawbacks such as lesser data integrity, data confidentiality rate, authentication accuracy, failed to enhance the security. In order to overcome the issue, the proposed IoT-RITSAMMMDC technique is presented. Followed by, the Ruzicka similarity indexive two-step authentication technique is applied to improve the accuracy. Merkle–Damgård Matyas–Meyer Message-Digest Cryptography is employed to achieve the higher security for avoiding authorized users. The major contributions of the proposed IoT-RITSAMMMDC technique are summarized as given below,

- A novel IoT-RITSAMMMDC technique is introduced for secure data storage in cloud based on authentication and Cryptographic hash generation.
- Proposed IoT-RITSAMMMDC technique uses Ruzicka similarity indexive two-step authentication technique to carried two-step authentication process.
- The novelty of Ruzicka similarity coefficient is used to verify the cloud user ID and password before the data storage. The similarity function provides the value between 0 to 1. When the value is 1, then password and ID are accurately matched. Then, the user is authorized user. When the value is 0, then password and ID is precisely not matched. Then, the user is an unauthorized user. Therefore, the accuracy is enhanced.
- To increase data confidentiality and data integrity, an IoT-RITSAMMMDC technique uses the Merkle–Damgård Matyas–Meyer Message-Digest Cryptography to generate the hash value for each authorized user's data and it stored on the server.
- The novelty of the Matyas–Meyer–Oseas compression function is employed with Merkle–Damgård tree to create the hash value in all industrial data. Then the hash value is stored in the cloud server. The novelty of the Cryptographic hash function is used to avoid unauthorized data access and modification to improve the data storage security.
- Finally, extensive experiments are conducted to evaluate the performance of our IoT-RITSAMMMDC technique and related works. The experimental result demonstrates that our IoT-RITSAMMMDC technique outperforms well than the other methods.

The rest of this paper is organized into five various sections. Section 2 introduces the related works. Section 3 describes the proposed IoT-RITSAMMMDC with different sub-processes. In Section 4, experiments are conducted with the dataset to illustrate the performance of the IoT-RITSAMMMDC. Section 5 provides comparative results discussions of the different parameters with the help of a table or graphical representation. Finally, the last section ends the work with the conclusion.

2. RELATED WORKS

A secured storage method was introduced in [11] for preserving the data. But it failed to apply the cryptography hash technique to guarantee secured storage. A remote data integrity auditing method was developed in [12] for secure data storage and hiding the sensitive data. Though the method reduces the computation overhead, higher data confidentiality was not achieved.

A Chinese Remainder Theorem (CRT)-based data storage method was introduced in [13] for storing the multiple user data securely on the cloud database. A novel cloud storage approach was introduced in [14] based on the combination of Ethereum blockchain and ciphertext-policy attribute-based encryption (CP-ABE). But the approach failed to ensure the data integrity to upload the document.

A secure identity-based aggregate signature (SIBAS) was introduced in [15] to attain data integrity. But it failed to perform the mutual authentication. A novel integrity verification approach was developed in [16] for securing data depends on Ternary Hash Tree (THT) and Replica based Ternary Hash Tree (R-THT). An efficient and secure big data storage system was developed in [17]. However, data confidentiality and integrity verification were not performed.

A secure deduplication method was introduced in [18] based on key sharing with lesser storage overhead. An enhanced secure threshold data deduplication method was developed in [19] for secure cloud storage. An outsourced dynamic provable data possession (ODPDP) method was introduced in [20] for secure data storage. But the method failed to use the privacy-preserving technique to ensure better security.

3. PROPOSAL METHODOLOGY

Cloud computing is a novel computational paradigm that offers various services to the users due to its performance, high availability, and low cost. Cloud computing provides the instantaneous storage services for the large data generated from the IoT devices. IoT devices are computing devices that connect wirelessly to a network and collect information. This information is stored on the cloud server and retrieved at anytime and anywhere through the web. But, still, several industries are not ready to implement cloud computing technology due to a lack of proper security control policy and limitation in protection. Based on this motivation, a novel IoT-RITSAMMMDC technique is introduced to prevent authorized access and also provide security of the data stored in a cloud server.

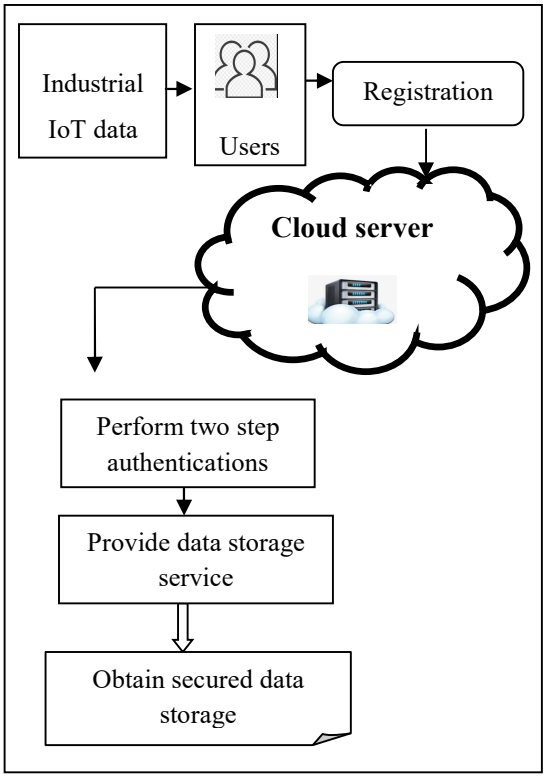

Figure 1: Architecture Diagram Of Proposed Iot-RITSAMMMDC

Figure 1 given above illustrates the architecture diagram of the proposed IoT-RITSAMMMDC technique to provide secure data storage with lesser space complexity in the cloud server. The network architecture comprises two entities such as cloud users $cu_1, cu_2, cu_3 \ldots\ldots cu_n$ who has the industrial data $d_1, d_2, d_3, \ldots. d_n$ to be stored in the cloud server 'cs' who provides the cloud storage services in a secured manner. The proposed technique

consists of three major processes namely registration, authentication, and secure data storage. These processes are explained in the following subsections.

3.1 Registration Phase

The proposed IoT-RITSAMMMDC technique starts to perform the registration process before storing the data into the cloud server. Let us consider the industrial application to securely store the data into the server. When the user wants to store the data, they first need to fill the registration form which is provided by the cloud server. The registration form includes personal information about the user like the original name, middle name, last name, date of birth, gender, mobile number, e-mail ID, and so on.

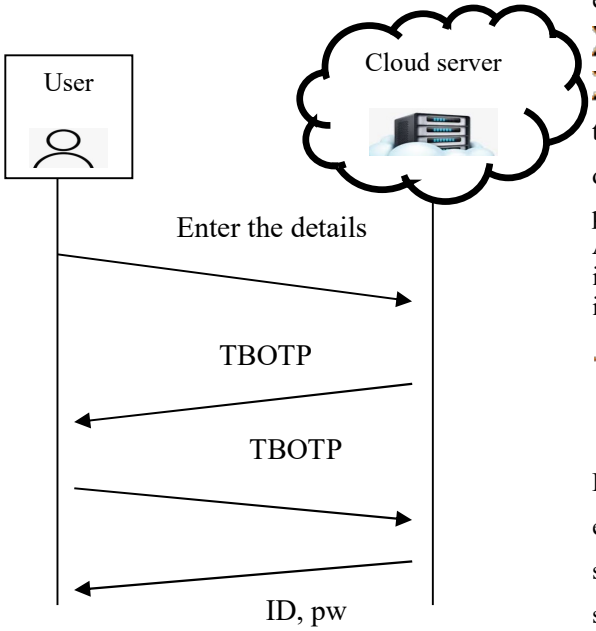

Figure 2: Flow Process Of The User Registration

Figure 2 displays the registration process of the user. The cloud user enters their personal information. After entering the details, the user clicks the submit button. Then the service providers transmit a Time-based One-Time Pin to the registered mobile number. Consequently, the user enters the received pin within a specific time. The user did not enter the pin at a particular time, then the user has to log in to the system again and enters their details. After entering the one time pin at a specific time, the cloud server sent successfully registered messages to the particular user. After the registration, the cloud server generates the ID and password for each registered user. The generated ID and password are also stored in the cloud server.

3.2 Ruzicka Similarity Indexive Two-Step Authentication

Whenever the registered user wants to store the data on a cloud server, they first verify their authenticity. The registered user first login to the cloud server and the server verifies the ID and password generated at the time of the registration phase. The authentication process is done with the help of the Ruzicka similarity index. Initially, the user ID is verified in the first step.

$$S\ (ID) = \frac{(ID)_e \cap (ID)_s}{\Sigma(ID)_e + \Sigma(ID)_s - [(ID)_e \cap (ID)_s]} \quad (1)$$

From (1), $S(ID)$ indicates a Ruzicka similarity coefficient, $(ID)_e$ denotes entered ID, $(ID)_s$ denotes an already stored ID. $\Sigma(ID)_e$ denotes sum of $(ID)_e$ score, $\Sigma(ID)_s$ denotes sum of $(ID)_s$ score. From (1), the intersection symbol '$\cap$' designates a mutual dependence. The similarity coefficient $S(ID)$ provides the integer value in the range from 0 to 1. After that, the password authentication is performed in the second step with the same Ruzicka similarity index.

$$S\ (pw) = \frac{(pw)_e \cap (pw)_s}{\Sigma(pw)_e + \Sigma(pw)_s - [(pw)_e \cap (pw)_s]} \quad (2)$$

From (2), $S\ (pw)$ indicates a Ruzicka similarity coefficient, $(pw)_e$ denotes entered password, $(pw)_s$ denotes an already stored password. $\Sigma(pw)_e$ denotes sum of $(pw)_e$ score, $\Sigma(pw)_s$ denotes sum of $(pw)_s$ score. From (2), the intersection symbol '$\cap$' designates a mutual dependence. The similarity coefficient '$S\ (pw)$' provides the integer value in the range from 0 to 1. Based on the similarity value, the authenticated user is correctly identified with higher accuracy.

Figure 3 illustrates the flow process of two-step authentication. Whenever the registered user wants to store data, the user first enters the valid ID in the login window. Then the cloud server uses the Ruzicka similarity coefficient to verify that the entered ID is matched with the IDs stored on the server database at the time of registration. If the Ruzicka similarity coefficient returns '1', then the entered ID gets matched with the already stored ID. Otherwise, the user is said to be an unauthorized

Journal of Theoretical and Applied Information Technology
30th November 2021. Vol.99. No 22

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

user and the cloud server denied the services. After that, the user enters the valid password (pw) in the login window. Then the cloud server uses the Ruzicka similarity coefficient verifies the entered password is matched with the IDs. If the similarity coefficient returns '1', then the entered password gets matched with ID.

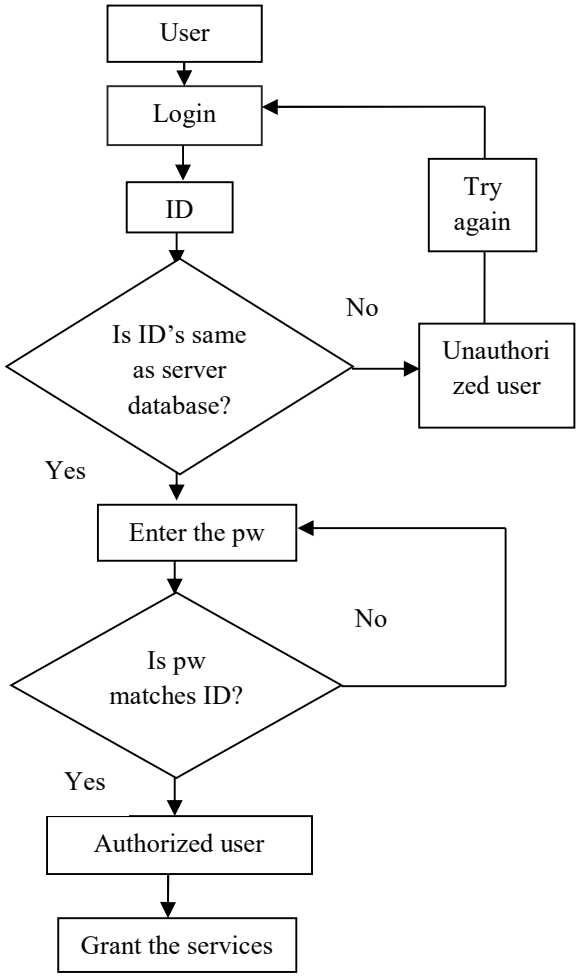

Figure 3: Flow Process Of Two-Step Authentication

Then the user is said to be an authorized user and the server grants the storage service.

3.3Merkle–Damgård Matyas–Meyer Message-Digest Cryptographic Hash-Based Secured Data Storage

After performing the authentication, secured data storage is carried out using Merkle–Damgård Matyas–Meyer Message-Digest Cryptographic hash function. The proposed technique uses the Message Digest (MD6) Cryptographic hash and it uses Merkle–Damgård structure to generate the hashes for very long input based on the Matyas–Meyer–Oseas compression function. The MD6 produces message digests of any desired size from 1 to 512 bits. The Merkle–Damgård is a tree structure that consists of a leaf node and a non-leaf node. The leaf node is tagged with the hash of a data block, and the non-leaf node is labeled with the cryptographic hash of its child nodes. The structure of the tree is illustrated in figure 4.

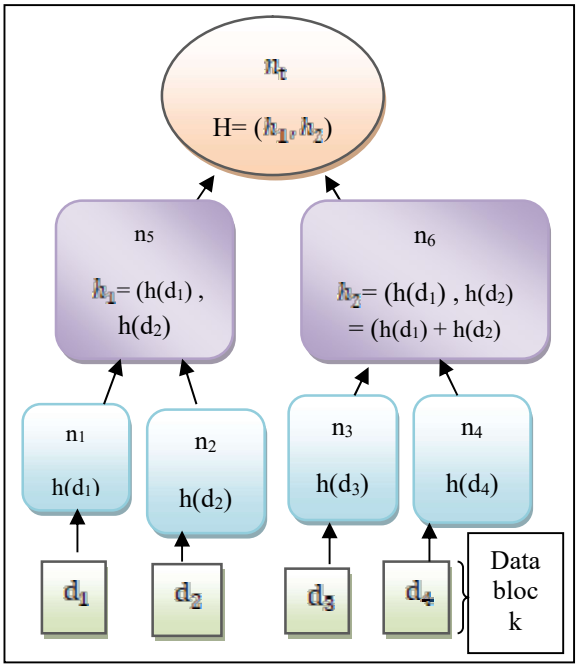

Figure 4: Merkle–Damgård Structures

Figure 4 illustrates the structure of the Merkle–Damgård tree which consists of the root node and leaf nodes and data block. In the tree, the data block has an industrial data $d_1, d_2, d_3, \ldots . d_n$ to be stored in the cloud server. The tree comprises the leaf nodes (n_1, n_2, n_3, n_4) labeled with the hash value of a data block $h(d_1)$, $h(d_2)$, $h(d_3)$, $h(d_4)$. The non-leaf node non-leaf nodes (n_5, n_6) are labeled with the concatenation hash value of its children nodes (n_1, n_2, n_3, n_4). The root node 'n_t' labeled with the concatenation hash value of its children nodes (n_5, n_6). Similarly, the industrial data are stored in the tree structure. The Merkle–Damgård tree uses the Matyas–Meyer–Oseas compression function to generate the hash for industrial data $d_1, d_2, d_3, \ldots . d_n$ in a data block.

The Matyas–Meyer–Oseas compression function is a function that transforms a two fixed-length of inputs into a fixed-length output. The

Journal of Theoretical and Applied Information Technology
30th November 2021. Vol.99. No 22

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

structure of the compression function is shown in figure 5.

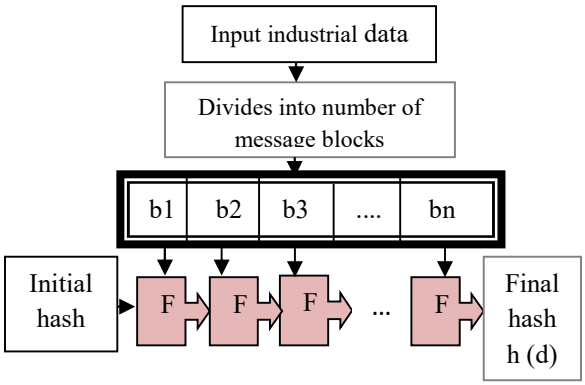

Figure 5: Matyas–Meyer–Oseas Compression

Figure 5 exhibits the Matyas–Meyer–Oseas compression function. Initially, the industrial data 'd' are given as input. The input data are partitioned into a number of message blocks with a fixed size.

$$d = b_1, b_2, b_3, \ldots b_n \quad (3)$$

From (3), $b_1, b_2, b_3, \ldots b_n$ denotes the number of blocks. The message block is given to the compression function (F) which takes input as a message block (b_1) and previous hash and finally generates the hash value (φ_h). The operation of the compression function is shown in figure 6.

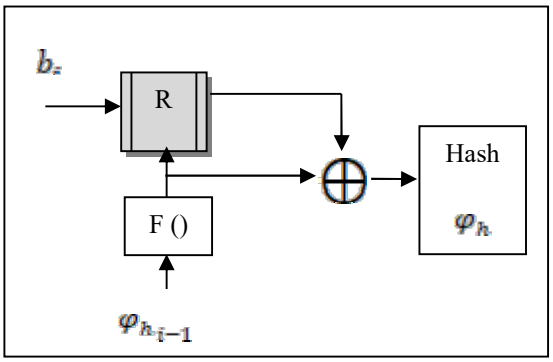

Figure 6 : Operation Of Matyas–Meyer–Oseas Compression Function

Figure 6 given above shows that the operation of the Matyas–Meyer–Oseas compression function which receives the message block 'b_i' and the previous hash value ($\varphi_{h_{i-1}}$) is preset. The previous hash value is fed into the function F () to be converted to fit as key for the block cipher 'R'. The output ciphertext 'c_d' is then XORed with the previous hash value and the message block (b_i) as the key to a block cipher 'R'. In the first round, the constant pre-specified initial hash value (φ_{h_0}). Therefore, the output of the compression function is expressed as follows,

$$\varphi_h = [R_{F(\varphi_{hi-1})}(b_i) \oplus b_i], \text{ Where } (i = 1,2,3, \ldots n) \quad (4)$$

From (4), φ_h denotes an output of the first compression function. The hash of one message block is not similar to another block i.e. $\varphi_{h_1} \neq \alpha_{h_2}$. The final hash of the data 'h (d)' is the output value of the last compression function. In this way, a hash value for each industrial data is generated and stored in the cloud server to avoid unauthorized access. This helps to increase the security of data storage in the cloud. The algorithmic process of the IoT-RITSAMMMDC technique is described as given below,

// Algorithm 1 IoT based Ruzicka index Two-Step Authenticative Matyas–Meyer Message-Digest Cryptography

Input: Number of users, $cu_1, cu_2, cu_3 \ldots\ldots cu_n$, number of industrial data $d_1, d_2, d_3, \ldots . d_n$

Output: Secured cloud storage

Begin

// Registration

Step 1: For each user cu_i

Step 2: cs ask to enter the details

Step 3: cu enters their details and send to 'cs'

Step 4: Server sends generates ID and pw

Step 5: end for

// Two-step authentication

Step 6: User login into the system with valid 'ID'

Step 7: cs verifies the ID using a similarity measure

Step 8: if $(S(ID) = 1)$ **then**

Step 9: Enter the password

Step 10: else

Step 11: User is said to be an unauthorized

Step 12: Denied the services

Step 13: end if

Step 14: if $(S(pw) = 1)$ **then**

Step 15: The user is said to be an authorized

Step 16: Cloud server allows to store the data

Step 17: **else**

Step 18: go to step 9
Step 19: end if
/// secured data storage
Step 20: Construct Merkle hash tree
Step 21: For each data 'd'
Step 22: Divided into a number of blocks $b_1, b_2, \ldots b_n$
Step 23: **For each** block b_i
Step 24: Generate **a** hash value φ_{h_i} using compression function 'F'
Step 25: end for
Step 26: Obtain final hash $h(d) = \varphi_{h_n}$
Step 27: Store hash $h(d)$ to cloud server
Step 28: end for
End

Algorithm 1 describes the step by step process of registration, authentication, and secured data storage. In the registration phase, the user sends their details to the cloud server. Accordingly, the server generates the ID and password for the user. Whenever the user wants to login into the system and stores the data, the cloud server first verifies their authenticity using the Ruzicka similarity function. If the password and ID are exactly matched, the similarity coefficient returns '1', and the user is said to be an authorized user. Otherwise, the user is said to be an unauthorized user. Then, the cloud server offers storage services to authorized users. This helps to improve security. Finally, the Merkle–Damgård Matyas–Meyer Message-Digest Cryptographic hash is applied to generate the hash value for each authorized user's data and it stored on the server. The Cryptographic hash function helps to increase data integrity and minimize the space complexity.

4. EXPERIMENTAL SETUP

Experimental evaluation of proposed IoT-RITSAMMMDC and existing SecACS [1], Multi-security-level cloud storage system [2] is implemented using Java language with CloudSim simulator. To conduct the experimentation, the Industrial Internet of Things Dataset is taken from the https://www.kaggle.com/pitasr/industrialiot for secure data storage on the cloud server. The cloud server provides the internet-based service that provides storage security for exchanging the energy-related Industrial Internet of Things information among the entities. The dataset comprises the industrial data collected using IoT devices. The dataset consists of 7 columns and 16382 instances. The 7 columns are DEMAND_RESPONSE, Area, Season, Energy, and Cost, pair no, Distance.

5. RESULTS AND DISCUSSION

The performance of the IoT-RITSAMMMDC technique over the existing methods namely existing SecACS [1], Multi-security-level cloud storage system [2] are discussed with various metrics are listed below,

- Authentication accuracy
- Data confidentiality rate
- Data integrity rate
- Space complexity

5.1 Authentication Accuracy

It is defined as the ratio of a number of users correctly authenticated as authorized or unauthorized to the total number of users in the cloud environment. The formula for calculating the authentication accuracy is expressed as given below,

$$AA = \left(\frac{Number\ of\ users\ correctly\ authenticated}{n}\right) * 100 \quad (5)$$

Where AA denotes an authentication accuracy, 'n' represents the number of cloud users. The authentication accuracy is measured in the unit of percentages (%).

Table 1: Comparison Of Authentication Accuracy

Number of cloud users	Authentication accuracy (%)		
	IoT-RITSAMMMDC	SecACS	Multi-security-level cloud storage system
25	88	84	80
50	90	86	82
75	92	88	84
100	90	86	83
125	92	88	85
150	91	87	83
175	93	86	84
200	91	88	83
225	93	89	85
250	92	87	85

Table 1 describes the comparison of the authentication accuracy for a number of cloud users. As shown in the tabulated results, the numbers of cloud users are taken in the counts from 25 to 250. The different results are observed for the various inputs. The observed results indicate that the proposed IoT-RITSAMMMDC technique increases authentication accuracy than th conventional methods. This is proved through the sample calculation performed with 25 users. By applying IoT-RITSAMMMDC technique, 22 users are correctly authenticated as authorized or unauthorized and the accuracy is 88%. Moreover, '21' users and 20 users are correctly authenticated as authorized or unauthorized and the accuracy are 84% and 80% using SecACS [1], a Multi-security-level cloud storage system [2]. From this result, it is inferred that the authentication accuracy is found to be comparatively higher using IoT-RITSAMMMDC when compared to [1] and [2]. Ten results are observed for each method. The overall results show that the authentication accuracy of IoT-RITSAMMMDC technique increased by 5% compared to SecACS [1] and 9% compared to Multi-security-level cloud storage system [2].

The observed results indicate that the authentication accuracy of the IoT-RITSAMMMDC technique is comparatively higher. This improvement is achieved due to the application of the Ruzicka similarity indexive two-step authentication technique. The cloud server uses the Ruzicka similarity coefficient to verify that the entered ID is matched with the IDs stored on the server database. Once the ID gets matched, then the password is verified. As a result, the verification of the password and ID using similarity function correctly authenticate the authorized or unauthorized users with higher accuracy.

5.2 Data Confidentiality Rate

It is measured as the ratio of a number of data is protected from the unauthorized user to the total number of data. Therefore, data confidentiality is measured using the following mathematical equation,

$$DCR = \left(\frac{Number\ of\ data\ protected}{total\ number\ of\ data}\right) * 100 \quad (6)$$

Where, DCR represents the data confidentiality rate which is measured in the unit of percentage (%).

Figure 7 portrays the experimental results of data confidentiality rate according to the number of data 1000 to 10000 taken from industrial IoT dataset. The observed results indicate that the IoT-RITSAMMMDC technique outperforms than other two methods.The average of comparison results shows that the data confidentiality rate of IoT-RITSAMMMDC technique is considerably increased by 6% when compared to [1] and 10% when compared to [2] respectively. The performance result of the data confidentiality rate is shown in figure 7.

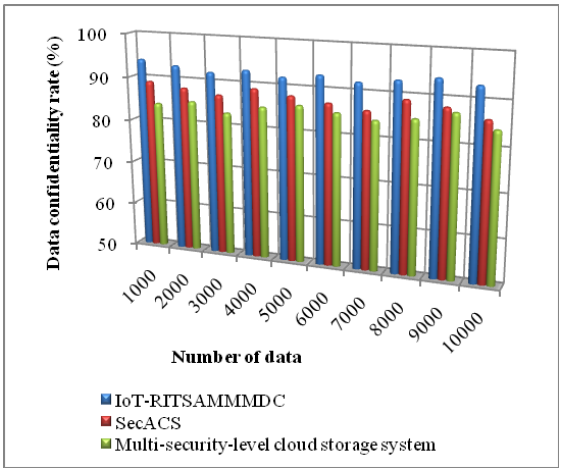

Figure7: Performance of data confidentiality rate

The performance of data confidentiality rate using different methods is exhibited in figure 7. The data confidentiality rate is measured as the number of data accessed by authorized users. From the above graphical illustration, it is obvious that the data confidentiality rate of three methods namely IoT-RITSAMMMDC technique, [1], [2] are represented by the three different colors of columns such as blue, red, and green respectively. The chart confirms that the IoT-RITSAMMMDC technique achieves a higher data confidentiality rate than the other two methods. This improvement of the IoT-RITSAMMMDC technique is attained through the application of the Merkle–Damgård Matyas–Meyer Message-Digest Cryptographic hash technology. The Matyas–Meyer compression function generates the hash value of each industrial data. Therefore, the hash values of data are accessed only by the authorized users and it avoids the unauthorized users. This process of the IoT-RITSAMMMDC technique achieves a higher data confidentiality rate.

5.3 Data Integrity Rate

It is another security parameter that referred to the number of data that are not altered by unauthorized users to the total number of data. **The** data integrity rate is measured as given below,

$$DIR = \left[\frac{Number\ of\ data\ not\ altered\ by\ unauthorized\ user}{Total\ number\ of\ data}\right] * 100 \quad (7)$$

Where DIR denotes a Data Integrity Rate which is measured in terms of percentage (%).

Table 2: Comparison Of Data Integrity Rate

Number Of Data	Data Integrity Rate (%)		
	Iot-RITSAMMMDC	Secacs	Multi-Security-Level Cloud Storage System
1000	93	87	83
2000	92	86	84
3000	91	85	82
4000	92	86	84
5000	91	87	85
6000	92	86	84
7000	91	85	83
8000	92	88	84
9000	93	87	86
10000	92	85	83

Table 2 indicates the performance of the data integrity rate with respect to the number of industrial data. Data integrity is defined as an amount of data that are not altered by authorized users to perform secure data storage on a cloud server. The observed results indicate that a variety of results are obtained for the different numbers of data. From the comparison results, it indicates that the IoT-RITSAMMMDC technique achieves a higher data integrity rate. As shown in the tabulated results, a totally ten different results are observed for each method. At the first run, the experiment is carried out with 1000 industrial data, 930 data are not altered by authorized users and the integrity rate is 93%. Besides, the 870 and 830 industrial data are not altered by authorized users and their integrity rates are 87% and 83% using [1] [2] respectively. Similarly, the various runs are performed and different results are obtained. Therefore, the overall data integrity rate of the IoT-RITSAMMMDC technique is compared to the other two methods. The average of ten results noticeably confirmed that the data integrity rate is significantly improved using IoT-RITSAMMMDC technique by 7% and 10% when compared to existing SecACS [1], Multi-security-level cloud storage system [2] respectively.

This is due to the application of the Matyas–Meyer compression function based on the Message-Digest Cryptographic technique. The compression function produces the fixed size of the hash with the fixed size of the input industrial data. Any modification of the input industrial data and it causes a severe change in the hash value. This helps to avoid the data modification by the unauthorized user resulting it increases the data integrity rate than the other methods.

5.4 Space Complexity

Space complexity is defined as the amount of memory space consumed for storing the data into the server. The formula for calculating the space complexity is given below,

$$SC = number\ of\ data * space\ (storing\ one\ data) \quad (8)$$

Where SC denotes a space complexity which is calculated in terms of megabytes (MB). Figure 8 demonstrates the space complexity using three different methods namely IoT-RITSAMMMDC technique, SecACS [1], Multi-security-level cloud storage system [2] represented by three colors of columns such as blue, red, and green.The result indicates that the complexity is found to be minimal using the IoT-RITSAMMMDC technique than the other two conventional storage systems. The quantitative analysis also proved that the proposed technique has a lesser storage space.

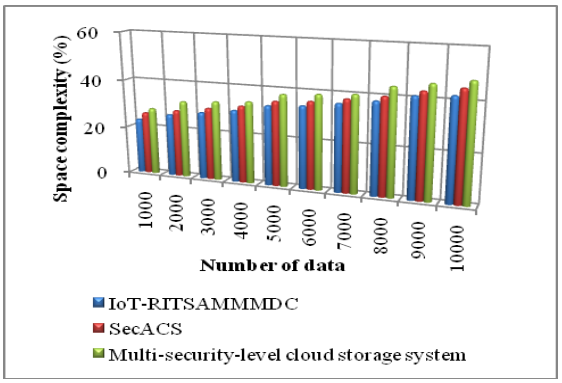

Figure 8: Performance of Space Complexity

This is due to the application of the Merkle–Damgård Matyas–Meyer Message-Digest Cryptographic technique. The authorized cloud user sends their data into the cloud server for storage. The cloud server uses the Matyas–Meyer compression function to generate the hash for each industrial data before storage. Therefore, the hash value of the data consumed lesser space than the original data hence the IoT-RITSAMMMDC technique minimizes the space complexity. The average of ten results indicates the space complexity of the IoT-RITSAMMMDC technique is considerably minimized by 6% and 14% when compared to existing [1] and [2] respectively.

5.5 Comparison with previous literature

From the experimental comparisons with the previous literature, the IoT-RITSAMMMDC technique provides better performance with higher data integrity. We compare the authentication accuracy, data confidentiality rate, data integrity rate, and space complexity of the IoT-RITSAMMMDC technique with the previous work. Comparison between IoT-RITSAMMMDC technique and previous literature is presented in Table 3 as given below.

Table 3: Comparison between IoT-RITSAMMMDC technique and previous literature

Method	Proposed system	Existing method [1]	Existing method [2]
	IoT-RITSAMMMDC technique	SecACS	Multi-security-level cloud storage system
Contribution	To enhance the security level for storing the data in cloud	To handle the data dynamic issue and store the data.	To preserve users' private data
Merits	Improved the confidentiality rate, integrity with less space complexity	Improved data integrity	To solve the data security issue with lesser time
Demerits	Parameter of throughput is not considered.	Data confidentiality rate was not enhanced.	The data confidentiality rate was not stored in cloud.
Authentication accuracy (%)	91.2%	86.9%	83.4%
Data Confidentiality Rate (%)	92.9%	87.7%	84.8%
Data Integrity Rate (%)	91.9%	86.2%	83.9%
Space complexity (ms)	33.1ms	35.3ms	38.2 ms

6. CONCLUSION

In this paper, proposed IoT-RITSAMMMDC is introduced to guarantee the security of data stored in the cloud server. It is designed with the application of the Ruzicka similarity index and Matyas–Meyer Message-Digest Cryptography. The contribution of the proposed technique is achieved to enhance the confidentiality rate, integrity, and minimum space complexity for storing industrial data.

The proposed IoT-RITSAMMMDC technique performs the registration process for entering their details and is stored into the cloud server. Then the server generates the ID and password for each registered user. Then, the proposed technique uses Ruzicka similarity Index to perform the two-step authentication. This helps to efficiently authenticate users with higher accuracy. The proposed technique applies the Merkle–Damgård Matyas–Meyer Message-Digest Cryptographic hash to store the data in the form of the fixed size of the hash value with lesser space complexity. The resultant hash of the data is protected and only accessed by the authorized user hence it improves the data confidentiality and integrity.

Finally, the proposed IoT-RITSAMMMDC technique is implemented using the Industrial IoT

dataset. Then, the experimental result demonstrated the validity of our proposed technique against the existing techniques. The results showed that the IoT-RITSAMMMDC technique mechanism provides better performance with an improvement of authentication accuracy, data confidentiality rate, integrity, and minimizing the computational time as well as space complexity when compared to the state-of-the-art works.

REFERENCES

[1] L.Li and J.Liu, "SecACS: Enabling lightweight secure auditable cloud storage with data dynamics", *Journal of Information Security and Applications, Elsevier*, vol.54, 2020, pp. 1-10

[2] J.Shen, X. Deng & Z. Xu, "Multi-security-level cloud storage system based on improved proxy re-encryption" *EURASIP Journal on Wireless Communications and Networking, Springer,* vol. 2019, 2019, pp.1-12

[3] F. Zhu, W. Wu, Y. Zhang, and X. Chen, "Privacy-preserving authentication for general directed graphs in industrial IoT", *Information Sciences, Elsevier*, vol. 502, 2019, pp. 218–228

[4] A. Esfahani, G. Mantas, R. Matischek, F. B. Saghezchi, J. Rodriguez, A. Bicaku, S. Maksuti, M. Tauber, C. Schmittner, and J.Bastos, "A Lightweight Authentication Mechanism for M2M Communications in Industrial IoT Environment", *IEEE Internet of Things Journal,* vol. 6, no. 1, 2019, pp. 288 - 296

[5] K. Grevenitis, F. Psarommatis, A. Reina, W. Xu, I. Tourkogiorgis, J. Milenkovic, J. Cassina and D. Kiritsis, "A hybrid framework for industrial data storage and exploitation", *Procedia CIRP, Elsevier,* vol. 81, 2019, pp.892-897

[6] C.Hu, Y.Xu, P.Liu, J. Yu, S. Guo, M. Zhao, "Enabling Cloud Storage Auditing with Key-Exposure Resilience under Continual Key-Leakage", *Information Sciences, Elsevier*, vol. 520, 2020, pp. 15-30

[7] B. P. Kavin, S.Ganapathy, U. Kanimozhi & A. Kannan, "An Enhanced Security Framework for Secured Data Storage and Communications in Cloud Using ECC, *Access Control and LDSA", Wireless Personal Communications, Springer,* vol.115, 2020, pp. 1107-1135

[8] B. Seth, S. Dalal , V. Jaglan, D. Le, S. Mohan, G.Srivastava, "Integrating encryption techniques for secure data storage in the cloud", Tra*nsaction on Emerging Telecommunications Technology, Wiley,* 2020, pp. 1-24

[9] V. Pavani, P. S. Krishna, A. P.Gopi, V. L.Narayana, "Secure data storage and accessing in cloud computing using enhanced group based cryptography mechanism", *Materials Today: Proceedings, Elsevier,* 2020, pp. 1-5

[10] L. Chen, J. Li, Y. Lu, Y.Zhang, "Adaptively Secure Certificate-based Broadcast Encryption and Its Application to Cloud Storage Service", *Information Sciences, Elsevier*, vol.538, 2020, pp. 273-289

[11] M. J. Reena and A. S. Nargunam, "Secured Storage of Big Data in Cloud", *International Journal of Recent Technology and Engineering (IJRTE),* vol. 8, no. 2S3, 2019, pp. 6-10

[12]W. Shen, J. Qin, J. Yu, R. Hao, J. Hu, "Enabling Identity-Based Integrity Auditing and Data Sharing With Sensitive Information Hiding for Secure Cloud Storage*", IEEE Transactions on Information Forensics and Security*, vol.14, no. 2, 2019, pp. 331 – 346

[13] B. P.kavin and, S. Ganapathy, "A secured storage and privacy-preserving model using CRT for providing security on cloud and IoT-based applications", *Computer Networks, Elsevier,* vol.151, 2019, pp.181-190

[14] S. Wang, X. Wang, Y. Zhang, "A Secure Cloud Storage Framework with Access Control Based on Blockchain", *IEEE Access*, vol. 7, 2019, pp. 112713 – 112725

[15] H. Li, L. Liu, C. Lan, C. Wang, H. Guo, "Lattice-Based Privacy-Preserving and Forward-Secure Cloud Storage Public Auditing Scheme", *IEEE Access,* vol.8, 2020, pp. 86797 – 86809

[16] M. Thangavel and P. Varalakshmi, "Enabling Ternary Hash Tree Based Integrity Verification for Secure Cloud Data Storage", *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no.12, 2020, pp. 2351 – 2362

[17] Y. Zhang, M. Yang, D. Zheng, P. Lang, A. Wu & C. Che, "Efficient and secure big data storage system with leakage resilience in cloud computing", *Soft Computing, Springer*, vol. 22, 2018, pp. 7763–7772

[18] L. Wang, B. Wang, W. Song, Z. Zhang, "A Key-Sharing Based Secure Deduplication Scheme in Cloud Storage", *Information Sciences, Elsevier*, vol. 504, 2019, pp. 48-60

[19] J. Stanek and L. Kencl, "Enhanced Secure Thresholded Data Deduplication Scheme for Cloud Storage", *IEEE Transactions on Dependable and Secure Computing ,* vol. 15, no. 4, 2018, pp.694 – 707

[20] W. Guo, H. Zhang, S. Qin, F.Gao, Z. Jin, W. Li, Q.Wen, "Outsourced dynamic provable data possession with batch update for secure cloud storage", *Future Generation Computer Systems, Elsevier,* vol. 95, 2019, pp. 309-322