Complex & Intelligent Systems (2021) 7:1733–1747
https://doi.org/10.1007/s40747-020-00162-3

ORIGINAL ARTICLE

A group-key-based sensitive attribute protection in cloud storage using modified random Fibonacci cryptography

M. Sumathi[1] · S. Sangeetha[1]

Received: 11 January 2020 / Accepted: 30 May 2020 / Published online: 16 June 2020

Abstract
Cloud computing is an eminent technology for providing a data storage facility with efficient storage, maintenance, management and remote backups. Hence, user data are shifted from customary storage to cloud storage. In this transfer, the sensitive attributes are also shifted to cloud storage with high-end security. Current security techniques are processed with high encryption time and provide identical security of entire data with single key dependent. These processes are taking high computational time and leaks entire information if the key is hacked. The proposed Group Key Based Attribute Encryption using Modified Random Fibonacci Cryptographic (MRFC) technique rectifies these issues. Instead of machine learning technique, data owner preference-based attributes segregation is used to divide an input dataset into sensitive and non-sensitive attribute groups. Based on inter-organization usage and data owner's willingness, sensitive attribute is divided into 'n + 1′ subgroups and each subgroup is encrypted by 'n + 1' group keys. The encrypted sensitive subgroups are merged with non-sensitive attributes and uploaded into a private cloud. The novelties of this paper are, (1) data owner preferred sensitive attribute classification instead of machine learning algorithms, (2) sensitive attribute encryption instead of entire attributes, (3) To reduce encryption time without compromising data owner privacy, (4) To decrypt and access the required subgroup instead of the entire attribute. Our experimental results show that, the proposed method takes minimal processing time, better classification accuracy and minimal memory space with high security to selected attributes as compared to existing classification and security techniques. Hence, sensitive data security and privacy is achieved with minimal processing cost.

Keywords Attribute group · Modified random Fibonacci cryptography · Group key · Sensitive attribute

Introduction

In cloud computing, Infrastructure as a Service (IaaS) offers reliable and scale effective storage services for storing a massive volume of data via the internet. IaaS reduces infrastructure costs and provides efficient management [1]. Nowadays, user data transactions are performed over the internet. These processes handle wide-ranging data with different genres and sensitivity. Traditionally user information was maintained on their personalized storage devices with their premises. This on-premise storage technique provided more protection than the third-party storage location-based cloud storage system. These cloud storage locations are controlled by third-party cloud service providers (CSP). Hence, cloud storage-based data falling under high-security issues. Ensuring security of cloud-based data is a crucial and critical process when it is used in financial and healthcare organizations. Because, these organizations are handling a large amount of sensitive information like the personal identification number (PIN), salary, disease, etc. Additionally, a large quantity of data is transferred into the online and storage of sensitive attributes in third party locations is increasing the probability of unauthorized access in the cloud storage system [2]. Nowadays, user information is retrieved by variety of people for refining their business, doing research work, and provides improved services to users. Thus, data usability is to be provided to inter-organization members and outside authorized users, without compromising data owner privacy is an essential task also. To achieve this requirement, sensitive attributes (S_A) are segregated from nonsensitive attribute

✉ M. Sumathi
sumathishanjai.nitt@gmail.com

S. Sangeetha
sangeetha@nitt.edu

[1] Department of Computer Applications, National Institute of Technology, Tiruchirappalli, Tamilnadu, India

(N_{SA}) with a knowledge of the data owner (D_O) is proposed in this work.

Generally, data classification techniques are used for data classification and symmetric or asymmetric key encryption algorithms are being used for data protection. These techniques are not suitable for a cloud-based storage system because the classification technique's accuracy depends on the training set and the attribute identification count is common all users. Similarly, conventionally symmetric/asymmetric encryption techniques are suitable to on-premises storage locations not for the cloud storage system. Hence, attribute-based encryption (ABE) techniques are preferred in cloud-based secure data storage. To provide efficient access control and protection to user information, different kind of ABE encryption techniques are used in cloud-based data. Those are key-policy attribute-based encryption, ciphertext policy attribute-based encryption technique, role-based attribute encryption technique etc. But, the security strength of the ABE depends on the number of attributes, key values or roles are involved in an encryption process [3]. Hence, the vertical and horizontal partition-based selective attribute encryption techniques are used for providing better security to selected attributes instead of entire attributes. Similarly, D_O's are willing to share their data with, authorized users for utilizing better services from them. This willingness is differing from user to user. Hence, alternate classification and security technique are required instead of machine learning and traditional security techniques for providing D_O-based security preferences to their data [4].

The main intention of encryption technique is to afford a proficient data access control, confidentiality, and integrity to S_A. However, few issues caused in current security techniques, such as higher encryption time, identical level security of entire attributes, single key-dependent encryption/decryption, non-involvement of D_O, and inter-organization members. These issues lead to critical problems and leak entire information if the key is hacked. The current security techniques like, Rivest–Shamir–dleman (RSA) has taken exponential encryption time for processing a smaller volume of data [5]. Message-Digest_5 (MD5) increases an encryption time and processed by the advanced encryption standard (AES) encryption technique. The AES encryption technique security strength depends on key secrecy. In the two-factor data security mechanism, data can be encrypted by a sender with knowledge of user identity and a secret key is used for accessing data. The foremost issues of the cloud-based data storage system are data security, management, and monitoring. Because, in cloud-based data storage, user information is maintained by cloud service providers. Hence, cloud service providers are having a complete control over user data. Thus, the data are not in D_O control and their data is accessed by all without their knowledge [6]. Hence, D_O control-based data storage and security are proposed in this manuscript.

The drawbacks of existing classification and security techniques are:

- The ABE technique provides lesser data confidentiality, monitoring, and access control to S_A in cloud storage.
- Higher computational cost and process overhead based on entire attributes encryption and decryption process.
- Non-involvement of D_O and inter-organization members.
- Possible to access S_A by inside adversaries.

Nowadays, partition-based security is an emerging technique for providing perfect protection to attributes with the level of sensitivity and secrecy. Similarly, some attributes do not require protection. If security techniques are applied to the entire attribute leads to processing overhead of an authorized user. Hence, user attributes need to be partitioned as S_A and N_{SA} and different kinds of security technique are applied to S_A is an essential task. Hence, data classification techniques are applied to user data. The existing data classification technique's accuracy depends on the training set. Thus, the D_O preference-based S_A segregation technique had proposed my previous research work [7]. The proposed MRFC algorithm is used to overcome this problem.

The major contributions and novelties of this paper are,

- To generate group key using a MRFC-ECC technique for providing higher security to sensitive attribute groups.
- The D_O and Group Admin (G_A) preference-based attribute encryption.
- To reduce encryption time with reduced key management and inside adversaries' harms.

The remaining portion of this manuscript is organized as follows: The existing works related to S_A classification and protection techniques in cloud-based data is discussed in "Related works". The detailed description of the Proposed MRFC key generation, data encryption, and decryption process with its algorithm is discussed in "Proposed MRFC-based secure sensitive attribute storage system". "Result and discussions" presents an experimental result with a comparison of previous methods and proposed techniques. In "Security analysis", the mathematical and security analysis of MRFC is discussed. Finally, the manuscript is concluded with future work is discussed in "Conclusion".

Related works

Sensitive attribute classification and protection

This section analyses the existing classification and protection techniques related to S_A. Identifying S_A through machine learning techniques are not suitable for all domains. Fast distributed mining (FDM) is used to identify private subsets from the entire data. The K-nearest neighbour (KNN) classification technique is used for classifying S_A from N_{SA} and S_A was encrypted with the RSA algorithm. RSA algorithm takes exponential encryption time and security strength depends on prime factor values. Additionally, S_A accuracy depends on the training set of the KNN algorithm [8]. Furthermore, the fuzzy logic classifier is used for classifying organization data into top-secret, secret, confidential, and public data. The level of encryption is determined as high, medium, and low. The data security and classification accuracy depend on organization and time specifications [9]. The existing S_A classification accuracy depends on the type of classifier and training set, not a D_O. Protection to S_A by partitioning attributes into several chunks with semantic meaning and places an encrypted chunk into the separate cloud. It reduces data usage to the authorized user and increases the computational complexity of merging into original information [10]. The S_A is classified from the N_{SA} by applying user-defined classification rules and S_A are encrypted with the AES algorithm. The S_A security depends on key transformation, key-size, and the number of rounds in an encryption process and classification rules [3]. This analysis clearly shows that the classification accuracy depends on a training set and the number of rules is involved in a classification process. Similarly, D_O is not involved in a classification process. Hence, the D_O preference-based SA classification technique is required in a current system.

Attribute-based encryption

This subsection furnishes the outline of existing ABE techniques. Cipher-text policy attribute-based encryption (CP-ABE) uses for the cooperative key management protocol to share data in the cloud. The storage of private and distributed key generation was added for immediate attribute revocation and fine-grained access was utilized to construct the private key update algorithm. This system was providing more security with high cipher-text size, inefficient access structure, and encryption/decryption cost [11, 12]. Weighted attribute-based encryption (WABE) method provides a fine-grained access control and was providing better performance than other schemes. The attribute weight was assigned by admin, not a D_O [13]. The dynamic search method for secure and efficient data access provides enhanced efficiency compared to a linear search with reduced access time and searching cost. The secure KNN algorithm was used to protect two threat models. Here, D_O was in-charge to create the updated data and a data was stored in a cloud storage location. The disadvantages were security challenges were occurred in the multi-user scheme and user revocation [14].

The decentralized access control technique is used for maintaining the data securely in the cloud. The CSP was verified the legitimacy of a server without the knowledge of user identity before storing data. This method was preventing replay attacks and supports the establishment, adjustment, and interpretation of data that was kept in a cloud storage location. An authentication and access control method were decentralized. The limitations were, access policy of each datum was kept in the cloud and was not concealing the attributes and access plan of the user. ABE was excellently permitted the users who were having access rights, is able to use a data security in the cloud. The D_O assist in key generation and management process. This method was taking the less computational time and reduced traffic burden with improved scalability. But the requirement of enough client infrastructures was uneconomical [15–17].

The sensitive, revocable, and proficient access control method for a multi-authority cloud storage technique achieved both forward and backward secrecy. Without D_O knowledge, the key generation and encryption processes are completed by organization members. The multi-authority storage technique was used in the remote storage system, online networks, etc. [18, 19]. Dual server public key encryption with keyword search (DS-PEKS) avoids inner keyword predicting attack which was an intrinsic weakness of the conventional PEKS framework. The smooth projective hash function (SPHF) was denoted as linear and homomorphic SPHF (LH-SPHF). The stronger security was achieved by the decision of Diffie–Hellman-based LH-SPHF [20]. The security protection is to be protecting the deployed data user privacy and providing data integrity, access control, and confidentiality of cloud data. The cloud security was improved their efficiency, reduced computational complexity, cost of bandwidth, and overhead in storage. However, it had drawbacks like difficult to maintain accountability, privacy protection, data integrity, and availability with low cost [21]. Role-based encryption (RBE) was integrated into the security technique by role-based access control (RBAC). The single key used for the decryption process and it operated efficiently irrespective of the role hierarchy and user membership complexity in the system [22]. Fully homomorphic encryption (FHE) technique was used for protecting a data and computation analysis. The data and computational analysis is divided into a number of subset and are encrypted by FHE technique which is maintained in separate cloud for providing an improved security to user data. The hard

clustering and fuzzy clustering algorithms were used for forming of data group. Each subgroup was encrypted separately by FHE technique [23].

Fibonacci cryptography

Quantum key distribution (QKD) protocol, used for the Fibonacci valued OAM entangled states. The Fibonacci matrix representation was well-defined to enhance the original protocol. It has not only enhanced the efficiency of encoding, but also verifiability. QKD protocol is used to attain the verifiability and this protocol is used for better implementation using recent technology [24]. The multiple variable factors and recovery of the original data were very difficult. The size of the circular queue had a tenable factor. The keyword letter and the numbers are denoted in the Fibonacci format. Using shift and logical operations, all letters are converted into ASCII binary format by security algorithm. This is mainly used for text messages. The results of the proposed algorithm had given a 50% lower complexity when compared to Multiple Circular Queue Algorithm (MCQA) [25].

Secure communication is established through the group key-based encryption task. The better encryption QKD was utilized with Lucas, Fibonacci, and Fibonacci–Lucas that gives the quantum signature verifications. This proposed technique improved the verification of signing and verified the information that is received from the participants for an authentication. This proposed final outcomes the protection by minimal delay when compared to a normal QKD technique [26, 27].

Proposed MRFC-based secure sensitive attribute storage system

Group-key-based sensitive attribute protection using Modified Random Fibonacci Cryptography (MRFC) in cloud storage system provides better security to D_O preferred S_A in a private cloud. The S_A and N_{SA} partition depends on privacy score values of individual attributes, where security preferences are given by D_O. Table 1 indicates the roles of each participant are involved in a proposed system.

D_O assign a Likert scale value (L_{SV}) to all attributes and the L_{SV} is converted to Dichotomous scale value (D_{SV}) for constructing response matrix $R_D(i, j)$ by general admin (G_{NA}). Sensitivity (β) and visibility ($V(i, j)$) values are calculated from $R_D(i, j)$. The threshold (T) is calculated from an average value of the privacy score value (P_{SV}). If P_{SV} of an attribute is lesser than the average privacy score value (threshold value), the specific attribute is partitioned as S_A otherwise attribute is partitioned as N_{SA} [7].

Now S_A is encrypted by the proposed technique and N_{SA} are stored as in a plain-text form in the cloud. Table 2 shows a list of symbols are used in the proposed MRFC technique. The current security techniques are depending on D_O preferences. Hence, data security fully depends on D_O not data handling organizations.

Nowadays, D_O perform their task through online and data transactions are performed between multiple organizations. When data is moved to inter-organizations, high security is required, to protect a S_A. The proposed method provides one such technique involves D_O, G_{NA}, Group Admin (G_A) [e.g. Insurance Admin (G_{A_1}), Marketing Admin (G_{A_2}), Loan Admin (G_{A_3})], and Cloud Service Provider (CSP). The D_O encrypts their Group of $S_A(G(S_A))$'s by a Group Key (G_K) before uploading into cloud storage. The G_A sends a request message to CSP for acquiring information about D_O for their process. The request is verified by CSP for authorization and sends the key request to D_O. Then, the request is analysed by D_O and send the encrypted $G(S_A)$ to the requested G_A.

Here, the G_K is generated by D_O is used for encryption of their $G(S_A)$. The major benefit of this technique is, without D_O approval, none of their data are shared with others and D_O will get the log records of the requester for every transaction from CSP. Hence, the D_O having complete access control and monitoring over their data. Now, the G_A can access the D_O data, depending on the decision provided by a G_{NA}. Figure 1 shows the proposed system architecture.

Table 2 Symbols used in proposed method

S. no	Symbols	Descriptions
1	D_O	Data Owner
2	G_{NA}	General Admin
3	G_A	Group Admin
4	S_A, N_{SA}	Sensitive Attribute, Non-Sensitive Attribute
5	G_K	Group key
6	P_r, P_u	Private key, Public key

Table 1 Roles of participants

S. no	Participants	Roles
1	Data owner (D_O)	Assign security level to each attribute and generate group key
2	Group Admin (G_A)	Access the required group of attributes and decrypt the attributes
3	General Admin (G_{NA})	Monitor access control of each user
4	Cloud service provider (CSP)	Maintain the encrypted S_A and non-encrypted N_{SA}

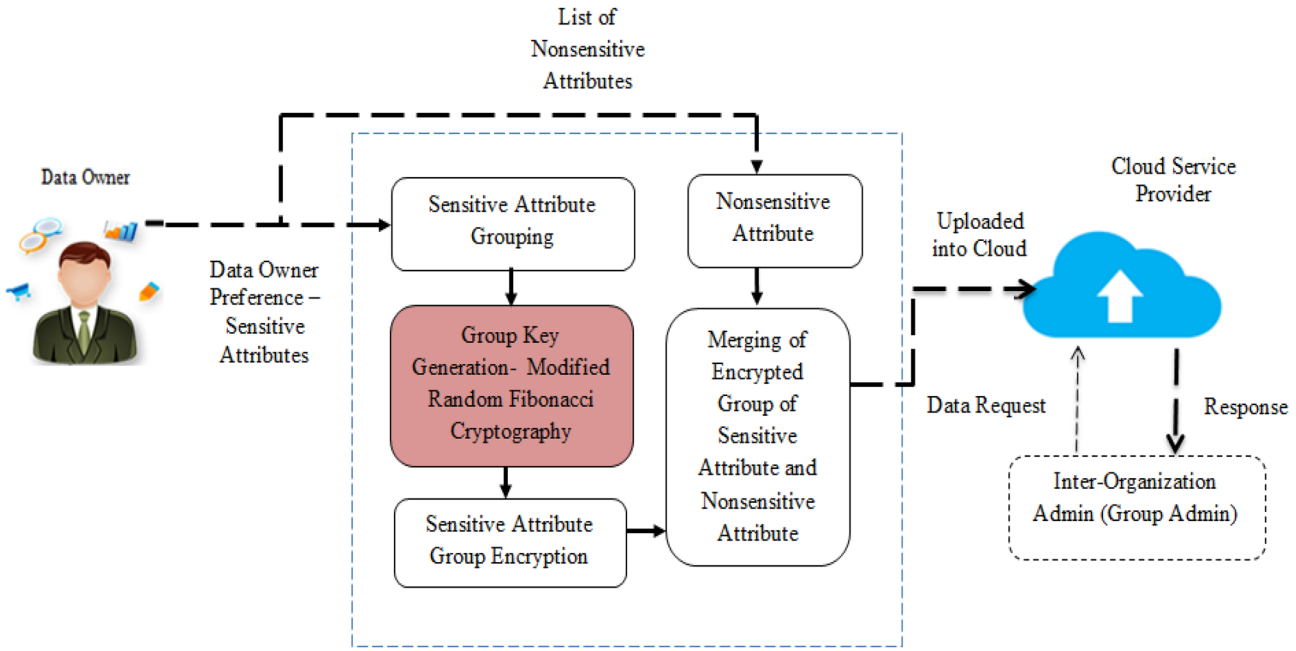

Fig. 1 System architecture for proposed MRFC

Modified Random Fibonacci Cryptographic Technique

The proposed MRFC technique describes the G_K generation process, S_A encryption/decryption, G_K sharing, and the encrypted $G(S_A)$ transfer. The flow depicts the files are uploaded with sensitivity preference given by D_O and the G_{NA} receives the sensitivity preferences. The G_{NA} analysed the security preference and splits(π) the data as S_A and N_{SA} such as attribute $\in \pi(S_A, NS_A)$. The identified S_A is grouped into 'n + 1' groups by G_{NA}. Similarly, G_K is generated by D_O using the MRFC technique. The generated G_K is used for encrypting the S_A and merged with N_{SA} for uploading into the cloud. The G_A sends a data request to CSP and the request is verified by CSP. If the G_A request is valid, the requested data are transferred to G_A by D_O.

Group key-based sensitive attribute protection

MRFC is established by an Elliptic Curve Cryptography technique that is used to generate keys over the properties of the Elliptic Curve equation "$y^2 = (x^3 + ax + b) \bmod p$". In MRFC, the key generation is combined with a Diffie–Hellman Key exchange technique for transfer a key between the parties.

Grouping of sensitive attributes

The groups are divided into three different categories listed as follows:

- S_A is the attributes are unknown to others/attribute required privacy.
- G_{A_i} is the attributes are accessed by ith GA.
- $C(G(S_A))$ is the $G(S_A)$ common to more than one G_A.

Algorithm 1 shows the S_A grouping process depends on the number of inter-organizations are going to access user information. The inter-organization requirement is gathered from the G_A which is, $R_{eq}(G_{A_i}) \in \{A_1, \ldots, A_n\}$. Then $R_{eq}(G_{A_i})$ belongs to either S_A or NS_A. Such as, if $A_1, \ldots, A_m \equiv S_A$ then $A_1, \ldots, A_m \in G_{A_i}(G(S_A))$ and $A_{m+1}, \ldots, A_n \in NS_A$, else $A_1, \ldots, A_m \in G_{A_{i+1}}(G(S_A))$. A similar process is repeated for the remaining G_A requirement-based S_A grouping. Here, some attributes are common for all. The common attributes $C(G(S_A))$ are identified as the intersection of all G_A requirements. The attributes which are subtracted from the $C(G(S_A))$ is identified as $G_{A_i}(G(S_A))$ or $G_{A_{i+1}}(G(S_A))$. Based on this process, two different groups of attributes are generated such as organization required attributes and D_O private attribute group. Now, the identified $G_{A_i}(G(S_A))$ is passed to the encryption process.

Algorithm 1: Grouping of Sensitive Attributes

Input: Sensitive Attributes (S_A), Group Admin (G_A)
Output: Group of Sensitive Attributes ($G(S_A)$)
Procedure

1. $A_1, \ldots\ldots\ldots, A_n \in G_{A_i}$
2. **If** $(A_1, \ldots\ldots\ldots A_m \equiv S_A)$ **then**

$$A_1, \ldots A_m \in G_{A_i}(G(S_A)) and\ A_{m+1}, .A_n \in NS_A$$

else

$$A_1, \ldots\ldots\ldots, A_m \in G_{A_{i+1}}(G(S_A))$$

3. Repeat this process for all G_A.
4. Identify the Common $S_A - C(G(S_A))$

if$(C(G(S_A)) \leftarrow G_{A_1}(G(S_A)) \cap \ldots\ldots\ldots \cap G_{A_N}(G(S_A)) \neq \emptyset)$

then

$$G_{A_1}(G(S_A)) = G_{A_1}(G(S_A)) - C(G(S_A)) \ldots\ldots$$
$$G_{A_N}(G(S_A)) = G_{A_N}(G(S_A)) - C(G(S_A))$$

else

$$G_{A_1}(G(S_A)) \ldots\ldots G_{A_N}(G(S_A))$$

5. **return** $[G_{A_1}(G(S_A)) \ldots\ldots G_{A_N}(G(S_A))\ and\ C(G(S_A))]$

Group key generation

The key generation is a process of producing G_Ks for the purpose of encryption. The key generation is the combination of Fibonacci values and Elliptic Curve Cryptography. Algorithm 3 shows the G_K generation using MRFC, taking an basic elliptic curve equation $y^2 = (x^3 + ax + b) \bmod p$ is for getting the initial and final values of random number generation. Based on these values, the initial parameters 'P' (initial value), 'Q' (Final values), and 'n' (number of values) is declared in the G_K generation task.

Cryptographic preliminaries *Bilinear Map* Consider a pair of cyclic groups (G, +) and (G,.) of a prime order P, and P_0 is a initiator of G. In a bilinear mapping process $e(GXG) \rightarrow G_1$ is true, and then it fulfils the given properties:

1. **Non-degeneracy:** $e(P_0, P_0) \neq 1$ is satisfied.
2. **Bilinear:** $e(P^x, P^y) = e(P^y, P^x) = e(P, P)^{xy}$ is true, for any $x, y \in F_P$ and $P \in G$.
3. **Computability:** The algorithm computes $e(P_1, P_2)$, for any $P_1, P_2 \in G$.

Hardness assumption of Decision Bilinear Diffie–Hellman Problem: Consider $\bar{A}$ be a polynomial time algorithm and it yields 'n' outputs (q, G, G_1, e). Here, 'q' is a prime number chosen based on 'n', G, G_1 and e. The decision bilinear Diffie–Hellman problem is hard (H) for any protection variables 'n', any probabilistic polynomial-time(PPT), distinguisher D, and any (q, G, G_1, e) generated by $\bar{A}(1^n)$, there is a negligible function *neg* is defined as

$$\left|\Pr(D(G, G_1, q, e, P^x, P^y, P^{Z,e}(P_0, P_0)^{xyz} = 1) - \Pr(D(G, G_1, q, e, P^x, P^y, P^{Z,e}(P_0, P_0)^{w} = 1)\right| \leq \text{neg}(n),$$

where P_0 is a random generator of G, and x, y, z and w are four identical components of F_P. The proposed system consists of the list of functions such as Setup, Group KeyGen, Encryption, Decryption and Adduser()/Revokeuser().

1. **Setup:** The G_{NA} generates a list of global parameters for the generation of (P_u, P_r) for each D_O and G_A. Elliptic Curve base point(G), rand(), EC points (P, Q) is taken as an input for G_K generation. Algorithm 3 describes this setup(). The user setup process, describes the list G_A and D_O are involved in a process. The G_A ID and D_O ID are assigned in this user setup process based on F_P. These ID's are stored in the cloud for D_O and G_A verification process by the cloud.
2. **Group KeyGen()**: Each D_O compute their (P_u, P_r) using MRFC technique. The Fibonacci values $\in \{P, Q\}$ is used for generating the P_r. Each D_O generate 'n + 1' G_K for encryption, if $G_{K_i} \in D_{O_i}$ who is registered in a setup phase is maintained their corresponding G_K.
3. **Encrypt**: $E(G_{A_i}(G(S_A))) \leftarrow Enc(G(S_A))$: Encryption algorithm run by the D_O to encrypt the group of S_A.
4. **Decrypt**: $D(G_{A_i}(G(S_A)))$—Decryption algorithm is executed by a G_A. $E(G_{A_i}(G(S_A)))$ and G_K are taken as an input and $D(G_{A_i}(G(S_A)))$ is produced as an output.
5. **Adduser()/Revokeuser()**: The adduser() and revokeuser() is executed by a D_O.

The security strength of the cryptographic algorithm is dependent on the random numbers. In a proposed technique, the modified Fibonacci cryptography is used for random number selection process. To solve the basic elliptic curve equation for the selection of initial and final positions of Fibonacci series. The generated Fibonacci values are taken as an input for the random number generation. Using Rand(), the random values are chosen from the Fibonacci values and are considered as a private key for each user instead of conventional elliptic curve-based private key selection process. This private key selection process increases the hardness of private key identification by an adversary. Because, an adversary unable to guess, which Fibonacci value is chosen as a private key for a particular group of sensitive attribute encryption. Using this technique, each D_O generates 'n + 1' private keys for their 'n + 1' group key generation. Thus, the proposed system provides high randomness than the conventional elliptic curve-based private key generation. In a proposed technique, each user chosen number of values for generating a group keys. Algorithm 2 shows the G_K generation process.

Algorithm 2: Group Key Generation

Input: Elliptic Curve (EC), Rand( )

Output: Group Key (G_{K_i})

Procedure:

1. Using an EC to fix an initial and final values of Fibonacci series
   $y^2 = (x^3 + ax + b) mod\ p$
2. $F[n] \leftarrow Fibonacci\_Series[P \ldots\ldots Q] \in EC.$
3. Initialize A [n] ← Rand(F[n])
4. **for(** i = 1 to n) **do**
   $P_{r_i} \leftarrow (Q * A[i] + P + \left(\frac{F[n]}{A[n]}\right))$
   end for
5. $K[n] \leftarrow P_{r_1} \ldots\ldots\ldots P_{r_n}$
6. $P_{u_i} \leftarrow P_{r_i} * G$
7. Return $G_{K_i} \leftarrow P_{u_A} * P_{r_B}$

In a proposed MRFC, the Fibonacci series $F[n] \in EC[P \ldots. Q]$. From $F[n]$, D_o use rand() to pick any one of the value and stored into $A[]$. Using this $A[]$ value, the P_r is generated for each D_O. Equation 1 is used for finding the P_r of a D_O:

$$P_{r_i} \leftarrow \left(Q * A[i] + P + \left(\frac{F[n]}{A[n]}\right)\right). \quad (1)$$

The generated 'P_r' values are stored into an array K[n]. The 'P_u' is calculated by multiplying 'P_r' with the base point 'G'. Then, $G_{K_i} \leftarrow P_{u_A} * P_{r_B}$. Similarly, the required number of G_{K_i} is generated and is used for encryption of $G(S_A)$. The key generation algorithm checks whether the $G_A \in O_i$ or not. If not, returns, $G_A \neq O_i$ else returns the G_K which consists of the tuples: $G_K \in (P, Q, A[i], F[i], P_r, P_u, B)$.

Sensitive attribute encryption algorithm

The $G(S_A)$ is encrypted with an appropriate G_K. A unique encryption scheme is adopted to encrypt a $G(S_A)$. Algorithm 3 shows the $G(S_A)$ encryption process. According to the number of organizations are available in a process the number of ciphertext (C) are generated.

Algorithm 3: Sensitive Attribute Encryption

Input: Group of S_A ($G(S_A)$), Group Key (G_{K_i})

Output: Encrypted Group of Sensitive Attribute ($E(G(S_A))$)

Procedure:

1. for all users i in a document
2. **if**($G(S_A)$) $\in G_{A_i}(G(S_A))$ **then**
   $$E\left(G_{A_i}(G(S_A))\right) = [\, G_{A_i}(G(S_A)) + G_{K_i}]$$
   else
   $$E\left(G_{A_N}(G(S_A))\right) = [\, G_{A_N}(G(S_A)) + G_{K_N}]$$
3. return $E\left(G_{A_i}(G(S_A))\right) \ldots\ldots E(G_{A_N}(G(S_A)))$

In an encryption process, if a $\left(G(S_A)\right) \in G_{A_i}(G(S_A))$, G_{K_i} is used for encryption. A similar process applies to the other $G(S_A)$ with different G_K. Now, the encrypted $G(S_A)$ is merged with NS_A and uploaded into cloud storage.

Sharing of group key

When a specific $G(S_A)$ is required, the P_u of D_O is used for finding the G_{K_i}. The 'P_u' is shared between D_O and G_A. Algorithm 4. shows the identification of G_K process.

Algorithm 4: Sharing of Group Key G_{K_i}

Input: $D_O(P_r, P_u)$, $G_A(P_r, P_u)$

Output: Group Key (G_{K_i})

Procedure:

1. Find the Group Key (G_{K_i}) from $G_A(P_u)$.
   $$G_{K_i} \leftarrow P_{r_A} * P_{u_B}$$
2. Find the Group Key (G_{Ki}) from D_O P_u.
   $$G_{K_i} \leftarrow P_{u_A} * P_{r_B}$$
3. **Return** G_{K_i}

When a G_A required for accessing a specific $G(S_A)$, their P_r is multiplied with the $D_O(P_u)$ for obtaining the G_K. The calculated G_K is used for decryption of the required $G(S_A)$.

Sensitive and non-sensitive attribute merging and transfer

The G_A sends a request to CSP, then the CSP verifies an authentication of G_A. If the verification is successful, the CSP sends encrypted $G(S_A)$ to G_A. The G_A decrypts the $G(S_A)$ using the corresponding G_K. The $G(S_A)$ and N_{SA} merging and transferring procedure is described in Algorithm 5.

Algorithm 5: Encrypted Sensitive Attribute Group Transfer

Input: Encrypted G(S_A), Non-Encrypted NS_A, Customer Account Number (C_{A_N})
Output: Transfer Requested G(S_A)
Procedure
1. **(i) Transfer Data to Cloud**
   for all i in E(G(S_A)) and for all j in N_{SA} do
   $C_A \leftarrow$ Merge(i, j)
   Transfer C_A to cloud
2. **(ii) Transfer request to Cloud**
   for all i in C_{A_N} do
   Transfer C_{A_N} from G_A to CSP
3. **(iii) Transfer C_A to G_A**
   if (i= =E($G_{A_i}(G(S_A))$) then transfer E($G_{A_i}(G(S_A))$)
   else if (i = = E($G_{A_{i+1}}(G(S_A))$) then transfer E($G_{A_{i+1}}(G(S_A))$)
   if (data= = E($G_{A_i}(G(S_A))$)then
   E(G_{K_i}) transfer to G_A
   else
   Transfer N_{SA} to G_A

The encrypted attribute $E(G(S_A))$ is completely secured in cloud-based sharing process. Afterwards, the approved G_A of the process are able to access E(G_{K_i}). In a proposed system, requested customer detail is sent to G_A instead of entire customers. This process takes lesser transfer time, decryption time, reduce unnecessary data transmission cost with higher security.

Sensitive attribute decryption

The S_A decryption process is an inverse function of the S_A encryption process. If a G_A wants to decrypt and access the $G(S_A)$, the User ID is verified whether the G_A is a non-revoked G_A or not. If the G_A is a non-revoked G_A, then CSP forward a request to D_O and D_O verifies their authenticity and send the requested $E(G(S_A))$. Algorithm 6 specifies the decryption process of the subgroup based on G_A request.

Algorithm 6: Sensitive Attribute Decryption

Input: Group Key (G_{K_i}), Encrypted G(S_A)
Output: Decrypted G(S_A) (D(G_A(S_A))
Procedure
1. **for** the request G(S_A)

$$D\left(G_{A_i}(S_A)\right) = \left[G_{A_i}(G(S_A)) + G_{K_i}\right] - \left[G_{K_i}\right]$$

   end for
2. Return D(G_{A_i} (S_A))

The specific requirement/application dependent attributes are decrypted from the data instead of complete data. The decryption task required a particular G_K's from the specified, G_A and D_O. If G_A revoked from an organization, they are unable to find the G_K. Since, the G_A related G_K is deleted from list during a revocation process. That is the revoked G_A's are unable to access $G(S_A)$ from a decryption process.

In decryption process, the G_A needs to calculate the G_{K_i} from their P_{r_i}:

$$P_{u_i} \leftarrow P_{r_i} * G,$$

If $P_{r_i} \in F_P$, then $P_{u_i} \in F_P$.
Now, $G_{K_i} \leftarrow P_{r_A} * P_{u_B}$.

If a requester is a revoked user then P_{r_i} and P_{u_i} not belongs to Fp. Thus, the revoked G_A's are unable to access the $G(S_A)$.

User revocation

When a G_A is revoked or added to process, user update, the ciphertext update, and G_K update are required in cloud storage for providing perfect security to S_A. Whenever the G_A revoked, the new G_K generates and perform encryption based on new G_K for the specific $G(S_A)$. The revocation system consists of the following process:

1. Delete the User ID of the G_A from the CSP and D_O in a dynamic manner.
2. Include the revoked G_A user ID in the revoked user list.
3. Choose a new random number (R^1) ∈ Fibonacci Series.
4. Compute new (P_r, P_u) key pair for the generation of G_K.
5. Now, new ciphertext is generated for the $G(S_A) \in$ revoked user and upload into the cloud.

Result and discussions

The results of the proposed MRFC algorithm are discussed in this segment. The created synthetic structured data for banking is used in the proposed method contains more sensitive information about the D_O. The data consists of 1000 records with 25 attributes are used in this process. From these 25 attributes, 15 attributes are S_A and 10 attributes are NS_A. This S_A and NS_A count are varied according to the user preference like user_1 $S_A = 15$, user_1 $S_A = 12$, user_3 $S_A = 8$, etc. The S_A is encrypted instead of complete attributes for reducing encryption time and computational complexity with high security. The proposed system is tested, and validation is done by JDK 1.7 in NetBeans 7.1. CloudMe is a private cloud that is used for data storage.

Sensitive attribute identification analysis

In a proposed system, the S_A is identified with the preferences of the D_O. The existing KNN classification technique-based S_A identification count depends on the training set. The attribute count given to the training set is 15. When an algorithm executed multiple times, the identified attribute count is same. But in a proposed system the identified attribute count varies depends on D_O preferences [7]. Figure 2

clearly expresses the S_A identification comparison of the proposed technique and the existing KNN classification technique.

Execution time analysis

In a proposed system the execution speed depends on the number of records is to be encrypted and decrypted. Instead of encrypting the entire attribute only S_A is to be encrypted and a specific group of attributes is to be decrypted instead of the entire attribute. This process takes lesser execution time than entire attribute encryption and decryption time. Table 3 shows the execution speed of the proposed system.

Encryption time analysis

An encryption time consumed by a proposed system with the various numbers of attribute group is shown in Fig. 3. The encryption time varies with the number of attributes. If the attributes count is less, the encryption and decryption time is less. When an attribute count is increased, encryption time is also increased. It clearly shows that the entire attribute encryption takes higher encryption and decryption time than the minimal sized attribute group. The MRFC encryption technique is more reliable than the conventional schemes and works well for larger volumes of data size. Because instead of entire data the partitioned group of attributes with the varies size is encrypted. Hence, the proposed technique takes lesser encryption time is proved. The encryption time is measured in milliseconds (ms). The time complexity is to be increased linearly when several groups are increasing.

In a proposed $G(S_A)$ storage technique, the G_A requirement is very less. Such as, each G_A required lesser than five S_A's. Each $G(S_A)$s are encrypted by separated G_K. Hence, the encryption time depends on the number of organizations are involved in a process. If 'n' organizations are involved in a encryption, then encryption time for entire $G(S_A)$ is defined as follows:

$$\text{Encryption time } G(S_A) = n \times \text{number of groups.}$$

Similarly, before encryption, the NS_A's are separated from S_A. These NS_A's are not involved in an encryption process. Hence, this proposed system encryption time is lesser than the entire attributes encryption time.

Decryption time analysis

Due to ECC-based key generation, the proposed MRFC technique is working faster than RSA-based encryption and decryption process. In the proposed technique, both the encryption and decryption depends on the G_K which is shared between D_O and G_A. Instead of decrypting entire attributes, the specific $G(S_A)$ is decrypted by a G_A. This decryption process requires minimal time than the complete data decryption process. Figure 4 provides the decryption time analysis of the proposed scheme.

This Fig. 4 clearly shows that the proposed technique takes lesser decryption time than the entire attribute decryption. This encryption and decryption time reduction process reduced the authorized user processing overhead.

Table 3 Execution speed

Number of attributes	Size of group (MB)	Encryption time (ms)	Decryption time (ms)
5	4.183	3.89	3.48
10	6.928	4.60	4.20
15	7.748	5.53	5.21
20	9.965	7.86	7.43
25	12.657	9.69	9.25

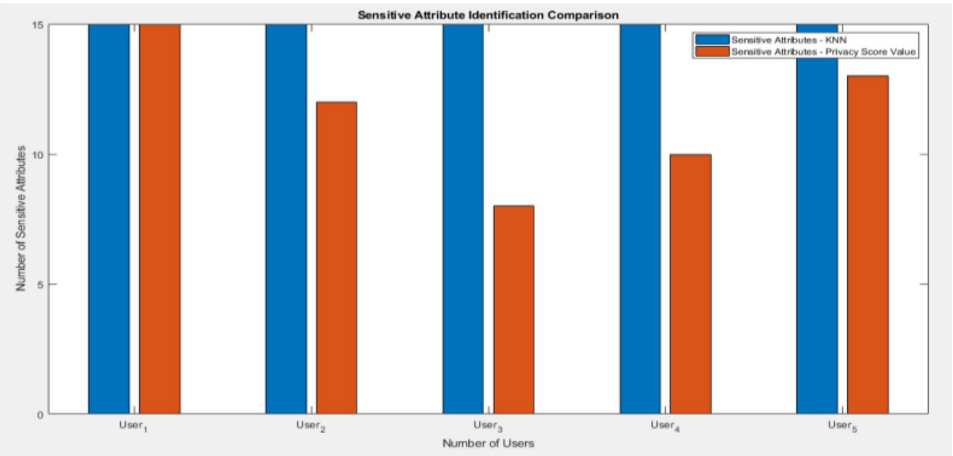

Fig. 2 Sensitive attribute identification comparison

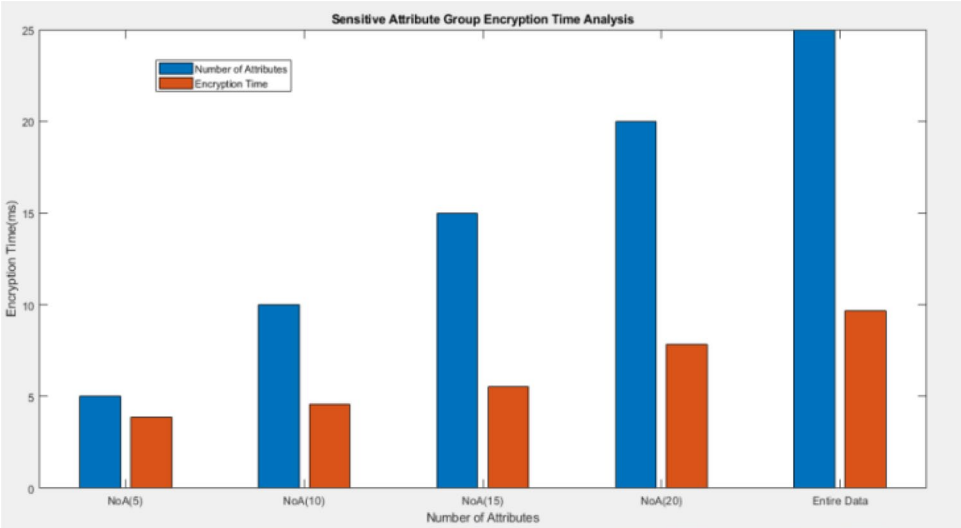

Fig. 3 Encryption time analysis

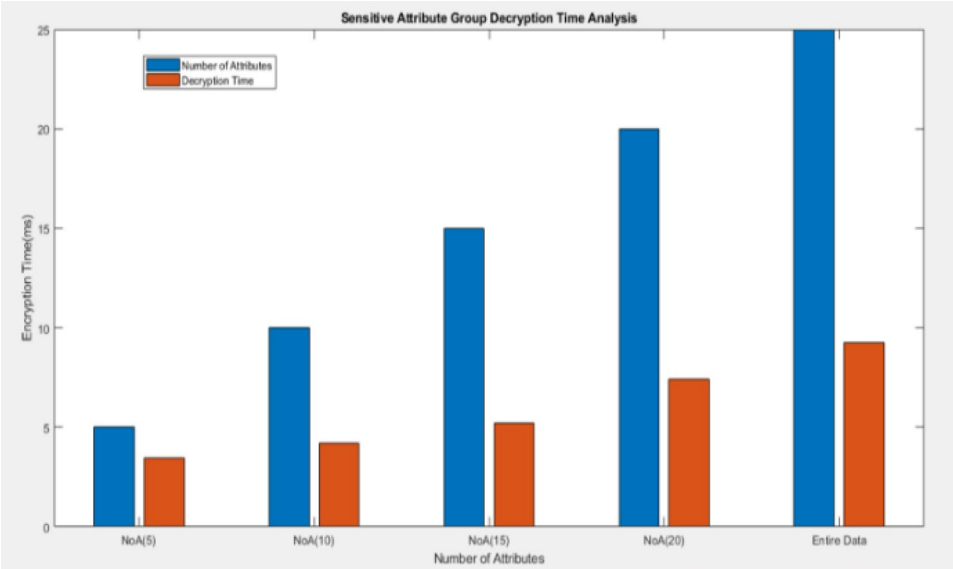

Fig. 4 Decryption time analysis

Memory space utilization analysis

Table 4 and Fig. 5 shows the memory space consumption of the proposed work in an encryption process. The memory space utilization is represented in the y-axis and the file size, that are used for experiments are represented on the x-axis. The amount of storage space is required to execute the algorithm with an input amount of data is known as encryption storage space. Here the proposed method occupied lesser storage space as compared to existing techniques. The memory space consumption is computed using the given formula:

Consumed memory space = total memory space − amount of free space

Less memory space utilization requires minimal storage cost in the cloud. Cloud computing is a pay per usage model, based on these characteristics the memory cost of S_A encryption takes minimal cost than entire data encryption.

Key generation time analysis

The key generation shows the number of keys used and compared to an author Rui Ruo work [28]. The proposed system has a lesser time. The key generation time for the proposed system is shown in Table 5. The key generation process is a

Table 4 Memory space consumption

Algorithm	Memory used (MB)
DES	18.2
3DES	20.7
AES	14.7
Blowfish	9.38
Proposed MRFC	8.83

one time process, hence, key updating overhead is not in a proposed system.

The proposed MRFC system has taken minimal time than the RUIGUO technique for different numbers of key generation. The number of key generation depends on the number of organizations are involved in a process. E.g. If four organizations are involved in a process, four G_K's are generated for encryption. A similar process is continued for other cases.

Through these experimental results the proposed D_O preference-based S_A identification technique satisfied the D_O requirement is proven. Similarly, the proposed MRFC-based S_A protection technique takes lesser encryption/decryption time, memory space consumption, key generation time than the entire attribute processing time is proven. The major role of the cryptographic technique is to provide secure data storage and communication. Hence, the security strength of the proposed system needs to be proved is a necessary task. The following section discussed the security strength of the proposed system.

Table 5 Key generation time analysis

No. of keys	RUIGUO (ms)	MRFC (ms)
4	0.15	0.06
5	0.91	0.36
6	1.68	1.39
7	5.87	3.2

Security analysis

The components that are used for a secure and efficient storage representatin is, the keyspace, security of data against attacks, computational speed, information entropy and correlation coefficient [29].

Keyspace analysis

The complete keys that are used in the cryptographic technique is known as keyspace. The strength of the technique is depending on the length of the key. If the keyspace is longer, the more resistant the algorithm is to a successful brute force attack. The key length is indicated by a number of bits. A *N*-bits (key length) has the keyspace 2^N possibilities. The size must be greater than 2^{100} to give high-level security from the cryptographic point of view [30]. The G_K length of the proposed encryption algorithm is 256-bits, hence the single G_K space is 2^{256} bits. In a proposed system 'n' number of organizations are involved in a process. Thus, '$2^{256*(n+1)}$' keyspace is used. This space is sufficient for reliable, practical usage and avoids brute force attacks.

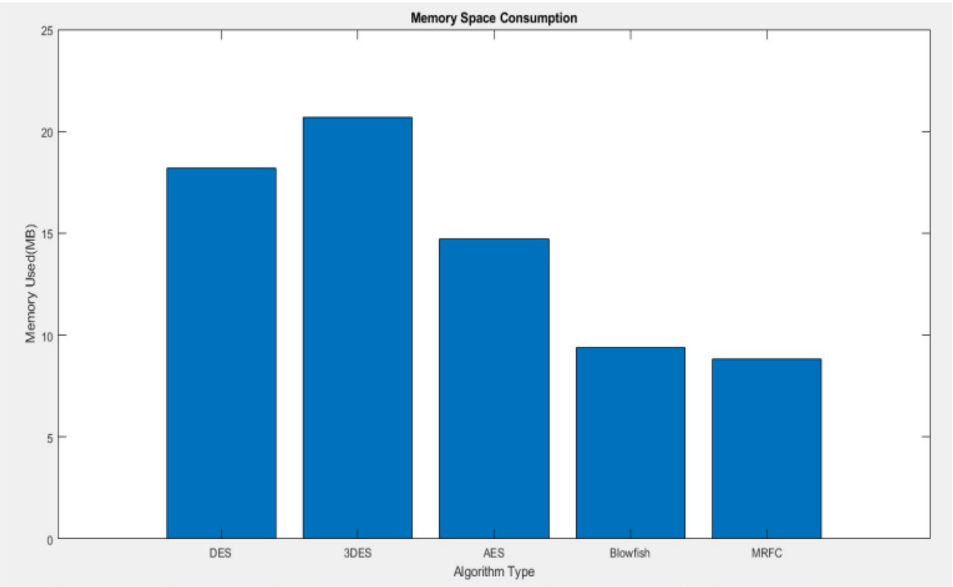

Fig. 5 Memory space consumption

Attack analysis

The well-known attacks are examined with the number of analyis steps and time requirement for a successful attack. Due to discrete logarithmic approach, the adversaries are unable to access the key or data in a polynomial time period. This is proven in the forthcoming points.

- *Inside attack* In a proposed system, the S_A is grouped by 'n + 1′ groups and every individual group is encrypted by an individual G_K by a D_O. After the encryption process, the encrypted $G(S_A)$ is uploaded into CSP. Each group is to be encrypted by a separate G_K; hence, no one can predict the G_K's which is used in the encryption of a specific group. Thus, the inside attacks are avoided in the proposed MRFC technique.
- *Outside attack* In a G_K-based access system, the $S_{A,}$ which is accessed by G_A is also possible by accessing other members involved in a specific organization. To overcome this drawback, the $G_A(P_r)$ is used for G_K generation. Hence, the P_r of G_A is required for the decryption of specific $G(S_A)$. Hence, no one can access group information.
- *Brute force attack* In a brute force attack, an adversary tries to identify the G_K's and plaintext messages in two different ways such as guessing and forging of G_K. In both cases, the adversary (Ã) tries to identify the key within a polynomial time. In a brute force attack, Ã tries all possibilities within a polynomial time period. But, in a proposed system the key length is $2^{\{256*(n+1)\}}$, to identify such larger key size in a specific time period is a complicated task. If an adversary identifies anyone G_K, the remaining G_A's are unable to predict. Because, each G_K is independent of the others and depends on D_O and G_A. E.g. Three G_A's are involved in a proposed system. Therefore, $2^{[256+256+256+256]} = 2^{1024}$ is the actual key size. Generally, 2^{256}-bit key size provides higher-level security in ECC than the 2^{1024}-bit RSA. In a proposed system, 2^{1024}-bit key is used. Hence, it is unbreakable in a polynomial-time period.
- *Known group key (resilience)***Theorem** *If any of the G_K is identified with an Ã, the remaining G_K cannot be found further. Because, each G_K belongs to independent MRFC-ECC-based random numbers.****Proof***: Based on MRFC-ECC-based G_K, the other G_K's should not be disclosed from the compromised G_K. Assume that, an Ã compromised anyone $G_{K_i} = P_{u_A} * P_{r_B}$, where $P_{u_A} = (P_{r_A} * G)$ and $P_{r_i} \leftarrow (Q * A[i] + P + \left(\frac{F[n]}{A[n]}\right))$. The security of the G_K depends on the randomness of P_r. If an Ã used the compromised G_K, to find other $P_r^1 \leftarrow (Q^1 * A[i+1] + P^1 + \left(\frac{F[n^1]}{A[n^1]}\right))$. The Ã knows the information of $G_K \in G(S_A)$ only. If an Ã used G_K^1, for accessing of other $G(S_A)$ is not possible due to ECDLP and Fibonacci series-based random function, the hardness of the proposed system is high. That is,

$$P_{r_i} \leftarrow \left(Q * A[i] + P + \left(\frac{F[n]}{A[n]}\right)\right)$$
$$\neq P_r^1 \leftarrow \left(Q^1 * A[i+1] + P^1 + \left(\frac{F[n^1]}{A[n^1]}\right)\right)$$

- That is, due to random function R, the $A[i]$ values are differing for each P_r and $G_K \neq G_K^1$. Similarly, the $R \in \{P, Q\}$ values. If the ECC prime value is high, the hardness of the P_r identification is high. Thus, the proposed MRFC-ECC resists the known G_K attack.
- *Key compromise impersonate*

Theorem *If an Ã revels the $D_O(P_r)$, only that $G(S_A)$ are accessed by them. It's impossible to compute the remaining P_r of the same D_O.*

Proof If an Ã knows the $D_O(P_r)$ and tries to access another P_r of the same D_O. The proposed MRFC-ECC algorithm resists this attack. Suppose, Ã knows G_A's P_u and send (G_{ID}, P_{u_B}) to D_O. Now, D_O compute (D_{ID}, P_{u_A}) to Ã. However, in order to derive other G_K, Ã must obtain the corresponding P_r for that G_K. Due to the difficulty of the ECDLP and MRFC-ECC-based P_r, the Ã is unable to derive new Pr. Thus, the proposed MRFC-ECC resists the key-compromise impersonate attack.

- *Chosen plaintext attack* The proposed G_K technique is against the chosen-plaintext attack and it is discussed with a security game between an Ã and the challenger (Ĉ). In a chosen plain-text attack, the 'Ã' gets a ciphertext for an arbitrary plain-text and tries to reveal all or part of the message from the ciphertext.

Theorem 2 *Within a polynomial-time period 'Ã' unable to crack the specific $G(S_A)$ against the G_K with a challenge access structure in the security game of Elliptic Curve Diffie–Hellman (ECDH) holds its assumption. This game is discussed as follows:*

Proof: Game Initialization and Query for phase 1 The 'Ã' chooses the defy access rights (Ŕ) and sends it to the Challenger (Ĉ). In a setup phase, the 'Ĉ' executes an algorithm for generating a G_K and sends a G_K to 'Ã'.

Challenge Now, 'Ã' selects two attribute groups $G_1(S_A)$ and $G_2(S_A)$ and sends it to the 'Ĉ'. The number of attributes and size of these two groups is the same. The 'Ĉ' receives these

groups and generates random bit value $\partial \in \{0,1\}$. Now, the ∂ value is used for encryption of groups by 'Ĉ'. The 'Ĉ' returns the (∂ = Enc ($G(S_A)$, Ŝ, P_r) to the 'Ã'.

Query Phase-2 The 'Ã' sends another request message to 'Ĉ' for finding a further G_K. Based on this request, the 'Ĉ' does the same job in phase-1.

Guess The 'Ã' should submit the guess $\partial^1 \in \{0,1\}$ for ∂. The 'Ã' wins the game when $\partial^1 = \partial$. The 'Ã' wins the game is defined as $(P_r \partial^1 = \partial] - 1/2)$.

The proposed G_K scheme is said to be more secure against the chosen plain-text attack if no probabilistic polynomial-time adversaries have a non-negligible advantage in the above game.

- *Forward and backward revocation* When a new G_A is added to a group, the new G_K is generated for that G_A. Now, the new G_K is used for the encryption process. Similarly, if any G_A is revoked from their role, the G_K based on that G_A is also revoked. In this process, only the specific group of attributes is re-encrypted instead of all groups. Thus, the forward and backward revocation takes lesser complexity than the existing forward and backward revocation process.

  Forward secrecy If any G_A joining to the process and try to access the $E(G(S_A))$, the proposed MRFC-ECC provides forward secrecy to the new G_A.

Proof The forward secrecy of a MRFC-ECC algorithm-based G_K is to all new G_A's to join in a process and tries to access D_O information; a new G_K is generated without modifying an existing group G_A's G_K. For generating a new G_K to a new G_A, the D_O check $G_A \in O_i$, if none of the existing G_A is not belongs O_i, and then new G_K is generated for G_A.

Backward Secrecy In cloud-based storage system, the user revocation and adding is a regular process. If the $G(S_A)$ is encrypted by a specific G_K, is needed to be updated.

Proof In a revocation process, a new random number R^1 is chosen for P_r generation:

$$G_K^1 \in \left(D_{O,} G\left(S_A\right), G, P_r^1, P_u^1, R^1\right)$$
$$G_K \neq G_K^1$$
$$E\left(G\left(S_A\right)\right)^1 \in G_K^1$$
$$E\left(G\left(S_A\right)\right) \in G_K i.e. E(G\left(S_A\right))^1 \neq E\left(G\left(S_A\right)\right).$$

In this analysis the ciphertext with $G_K \neq G_K^1$. Hence, the revoked G_A is unable to access the new G_K-based $E(G(S_A))$.

Mathematical proof

This section discussed the various comparative analyses in terms of security and storage overhead is discussed in Tables 6 and 7 respectively. Table 6 lists the various mathematical descriptions used for analysis.

Due to different G_K usage, the difficulty in the identification of each key is high. Hence, the proposed system is a collusion resistance (Co-Res) free, supports both backward and forward revocation process (B-F), provides confidentiality against CSP (Ag-Cloud), and user (Ag-User). Similarly, the proposed system, provides provable security, integrity, and access control system.

Table 7 shows the comparative security analysis for various existing techniques such as distributed access control scheme in cloud (DACC), Data access control- multi-authority cloud storage system (DAC-MACS), extensive data access control-multi-authority cloud storage system (EDAC-MACS) and proposed MRFC technique. These techniques can be compared in terms of collusion resistance, revocation security, data confidentiality, provable security, integrity and access control against the static corruption of authorities. Our proposed techniques obtained security additionally in integrity and access control compared to the other existing techniques. E.g. Each group of S_A is accessed by an individual organization through separate G_K. This G_K is generated by a D_O and the D_O having complete control over their data. Through this process, the access and integrity of the proposed system are maintained. Hence it is observed that our proposed technique has better security.

Table 8 shows the comparative analysis of the storage overhead for the existing DACC, DAC-MACS, NEDAC-MACS, and proposed approach. The existing techniques may have multiple attributes that need more storage. But our proposed MRFC technique does not contain multiple attributes. Only the minimal sized S_A is to be processed and stored, which reduces the storage overhead. As a result, our proposed technique improves performance with reduced storage overhead.

Table 6 Mathematical description

Notations	Description
$\|p\|$	Size of the element in group with a prime order p
t_c	Number of attributes associated with a ciphertext
n_c	Number of ciphertext in the cloud
t_u	Number of attributes of a user
n_{GA}	Number of attributes managed by G_A
N_{GA}	Number of G_A are involved in system

Table 7 Comparison of security analysis

Schemes	CO-Res	Revocation		Confidentiality		Pr Sec	Integrity	Access control
		B	F	Ag cloud	Ag User			
DACC	√	√	×	√	√	√	×	×
DAC-MACS	×	×	√	√	×	√	×	×
EDAC-MACS	√	√	√	√	√	√	×	×
MRFC	√	√	√	√	√	√	√	√

Table 8 Storage overhead analysis

Scheme	Group admin	Data owners	User	Cloud
DACC	$2n_{a,k}\|p\|$	$\left(n_c + 2\sum_{k=1}^{N_A} n_{a,k}\right)\|p\|$	$\left(n_{c,x} + \sum_{k=1}^{N_A} n_{a,k,uid}\right)\|p\|$	$(3t_c + 1)\|p\|$
DAC-MACS	$(n_{c,k} + 3)\|p\|$	$\left(3N_A + 1 + \sum_{k=1}^{N_A} n_{a,k}\right)\|p\|$	$\left(3N_A + 1 + \sum_{k=1}^{N_A} n_{a,k,uid}\right)\|p\|$	$(3t_c + 3)\|p\|$
NEDAC-MACS	$(n_{c,k} + 3 + n_u)\|p\|$	$\left(3N_A + 1 + \sum_{k=1}^{N_A} n_{a,k}\right)\|p\|$	$\left(2N_A + 1 + 2\sum_{k=1}^{N_A} n_{a,k,uid}\right)\|p\|$	$(3t_c + 3)\|p\|$
MRFC	N/A	$\left(N_A + 1 + \sum_{k=1}^{N_A} n_{A,k}\right)\|p\|$	$\left(N_A + 1 + \sum_{k=1}^{N_A} n_{a,k,uid}\right)\|p\|$	$(t_c + 1)\|p\|$

Conclusion

This paper proposed the sensitive characteristic-based encryption for the secure cloud storage system using MRFC. The MRFC provides enhanced security to sensitive data with nominal processing cost and security is provided through the data owner knowledge. The sensitive data are grouped into 'n + 1' groups and every individual group is encrypted by different G_K. Hence, to identify entire data from single key breaches is a difficult and impossible task. The encryption and decryption process is performed with the knowledge group admin. Hence, insider attacking is not possible. Similarly, the collision resistance, forward and backward revocation, chosen-ciphertext attacks, known-plaintext attacks are avoided through a group key-based encryption process. The key management problem is completely overcomes through the group key and is managed by individual data owners and group admins. The novelty of the proposed work is achieved through Group key-based encryption technique. The hardness of group key identification by an adversary is improved through MRFC-based elliptic curve technique. Usually, elliptic curve cryptography provides higher security with minimal key size. In addition to that, the random Fibonacci cryptography is used for the selection of random numbers which is used as a private key. This private key selection process improves the hardness of key identification by an adversary. Hence, the proposed technique overcomes the brute force attack, known group key attack, key compromise impersonate, chosen plaintext attack, and forward and backward user revocation processes in an efficient way. In future work, the same technique is going to be implemented in an unstructured complex document with images like a medical document.

References

1. Reddy VK, Surya KD, Praveen MS, Lokesh B, Vishal A, Akhil K (2016) Performance analysis of Load Balancing Algorithms in cloud computing environment. Indian J Sci Technol 9:1–7
2. Saini G, Sharma N (2014) Triple security of data in cloud computing. Int J Comput Sci Inf Technol 5:5825–5827
3. Alrawais A, Alhothaily A, Hu C, Xing X, Cheng X (2017) An attribute-based encryption scheme to secure fog communications. IEEE Access 5:9131–9138
4. Shimbre N, Deshpande P (2015) Enhancing distributed data storage security for cloud computing using TPA and AES algorithm. In: International conference on computing communication control and automation. IEEE, pp 35–39

5. Gupta NK (2018) Advancements in cloud computing software testing research. In: 4th int'l conf. on recent advances in information technology, RAIT-2018
6. Lin G, Hong H, Sun Z (2017) A collaborative key management protocol in ciphertext policy attribute-based encryption for cloud data sharing. IEEE Access 5:9464–9475
7. Sumathi M, Sangeetha S (2018) Scale-based secured sensitive data storage for banking services in cloud. Int J Electron Bus Inderscience Publ 14(2):2018
8. Xia Z, Wang X, Sun X, Wang Q (2015) A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. IEEE Trans Parallel Distrib Syst 27:340–352
9. Vyawhare PR, Ghuse ND (2020) Design and implementation of user anonymity and authentication scheme for decentralized access control in clouds: review. Int J Sci Res (IJSR) 3(11):1857–1861
10. Shabir MY, Iqbal A, Mahmood Z, Ghafoor A (2016) Analysis of classical encryption techniques in cloud computing. Tsinghua Sci Technol 21:102–113
11. Zhu Z, Jiang R (2015) A secure anti-collusion data sharing scheme for dynamic groups in the cloud. IEEE Trans Parallel Distrib Syst 27:40–50
12. Wei L, Zhu H, Cao Z, Dong X, Jia W, Chen Y et al (2014) Security and privacy for storage and computation in cloud computing. Inf Sci 258:371–386
13. Tysowski PK, Hasan MA (2013) Hybrid attribute-and re-encryption-based key management for secure and scalable mobile applications in clouds. IEEE Trans Cloud Comput 1:172–186
14. Yang K, Jia X (2013) Expressive, efficient, and revocable data access control for multi-authority cloud storage. IEEE Trans Parallel Distrib Syst 25:1735–1744
15. Ryan MD (2013) Cloud computing security: the scientific challenge, and a survey of solutions. J Syst Softw 86:2263–2268
16. Yang K, Jia X, Ren K, Zhang B, Xie R (2013) DAC-MACS: Effective data access control for multiauthority cloud storage systems. IEEE Trans Inf Forensics Secur 8:1790–1801
17. Chen R, Mu Y, Yang G, Guo F, Wang X (2015) Dual-server public-key encryption with keyword search for secure cloud storage. IEEE Trans Inf Forensics Secur 11:789–798
18. Tang J, Cui Y, Li Q, Ren K, Liu J, Buyya R (2016) Ensuring security and privacy preservation for cloud data services. ACM Comput Surv (CSUR) 49:13
19. Li Y, Gai K, Qiu L, Qiu M, Zhao H (2017) Intelligent cryptography approach for secure distributed big data storage in cloud computing. Inf Sci 387:103–115
20. Zhou L, Varadharajan V, Hitchens M (2013) Achieving secure role-based access control on encrypted data in cloud storage. IEEE Trans Inf Forensics Secur 8:1947–1960
21. Wang H (2014) Identity-based distributed provable data possession in multicloud storage. IEEE Trans Serv Comput 8:328–340
22. Sujithra M, Padmavathi G, Narayanan S (2015) Mobile device data security: a cryptographic approach by outsourcing mobile data to cloud. Procedia Comput Sci 47:480–485
23. Alabdulatif I, Khalil I, Yi X (2020) Towards secure big data analytic for cloud-enabled applications with fully homomorphic encryption. J Parallel Distrib Comput 137:192–204
24. Suneetha D, Kumar RK (2017) Data hiding using Fibonacci EDGE based steganography for cloud data. Int J Appl Eng Res 12:5565–5569
25. Albu-Rghaif AN, Jassim AK, Abboud AJ (2018) A data structure encryption algorithm based on circular queue to enhance data security. In: 1st international scientific conference of engineering sciences - 3rd scientific conference of engineering science (ISCES), pp 79–82, 10–11 Jan 2018
26. Raju UN, Vivekanandam R (2019) E-commerce security by quantum digital signature-based group key management. In: Innovations in computer science and engineering, LNNS, vol 74. Springer, New York, pp 251–262
27. Wahid MNA, Ali A, Esparham B, Marwan M (2018) A comparison of cryptographic algorithms: DES, 3DES, AES, RSA and Blowfish for guessing attacks prevention. J Comput Sci Appl Inf Technol 3:1–7
28. Guo R, Wen Q, Zhi H, Jin Z, Zhang H (2014) Certificateless public key encryption scheme with hybrid problems and its application to internet of things. Math Probl Eng 2014:980274
29. Mousa A, Faragallah OS, El-Rabaie S, Nigm E (2013) Security analysis of reverse encryption algorithm for databases. Int J Comput Appl 66
30. Mills D (2007) Review of cryptography: theory and practice by d. r.stinson. Cryptologia 31(1):87–88. https://doi.org/10.1080/01611190600964785