# Enhanced P-Gene based Data Hiding for Data Security in Cloud

**A Mallareddy, R Sridevi, Ch G V N Prasad**

***Abstract:**The advent of cloud computing has revolutionized the option of sharing cloud resources among the cloud users for minimizing the cost overhead.* ***The problem of data privacy and security needs to be addressed efficiently to fully utilize the power of cloud database services. Cloud data security is considered as a predominant issue that need to be addressed through the implementation of privacy preserving approaches that sustains and prevents the cloud resources and users from being compromised by the malicious intruders. Most of the proposed works for data hiding in cloud rely on the standard encryption techniques and steganography algorithms which have an overload of key distribution and secret key sharing. Hence these algorithms are not applicable to applications that require data hiding techniques with a faster computational time. In this regard, we propose an Enhanced Privacy preserving gene based data Aggregation Scheme (EPAS) for securing and exchanging private data by utilizing Enhanced P-Gene erasable data hiding approach. EPAS incorporates the benefits of the P-Gene which is responsible in the cloud space for providing security for the stored and private data in the cloud that are periodically exchanged with the clients of the cloud environment. Enhanced P-Gene scheme ensures secure sharing of data by relying on a trustworthy data aggregation scheme which is fully dependent on erasable data hiding technique.***

***Index Terms: Cloud Computing, Data Hiding, Data Security, Enhanced P-Gene.***

## I. INTRODUCTION

The advent of cloud computing has brought a dramatic change in the utilization of resources by considering them as a service such that the cloud user can access them from any place and any point of time [10, 15]. In cloud computing, the end users utilize the benefits of cloud resources without realizing its exact location such that optimal storage and access of data can be predominantly achieved [4, 8, 16]. This cloud environment also wide opens the feasibility of deploying and managing the cloud resources in order to prevent additional investment of capital with the focus for necessitating high potent network connectivity [9]. The cloud computing environment also enforces the periodic exchange of data that is independent of the location of data stored in the cloud infrastructure[17].

**Revised Manuscript Received on 30 May 2019.**

***** Correspondence Author
**A Mallareddy*,** AssociateProfessor, CVR College of Engineering and Research Scholar, JNTUH Hyderabad
**Dr. R Sridevi**, Professor and HOD, Department of CSE,JNTUH College of Engineering, Hyderabad
**Dr. Ch G V N Prasad**, Professor and HOD, Department of CSE, Sri Indu College of Engg & Tech., Hyderabad

The communication media used for transmission of data does not provide data security mechanisms which increases the risk of data misuse.The problem of data privacy and security needs to be addressed efficiently to fully utilize the power of cloud data services. In this context, data security is considered as the crucial issue that need to be concentrated in cloud data storage and transmission. Several solutions to address the security and privacy in cloud computing can be proposed as monitoring cloud server services, protection of data privacy, determination of the accuracy of the information, inaccessibility by third parties, protection against unwanted changes and deletions, prevention of malicious content and ensuring uninterrupted access to information[3]. However, majority of the security methods contributed for cloud storage incurred maximum overhead during the implementation of encryption with its inherent computational time complexity. Hence these algorithms are not applicable to applications that require data hiding techniques with a faster computational time.

We propose an Enhanced P-Gene technique for securing data stored in cloud and also during transmission. We particularly focus on a scenario wherein a group of user nodes, working for the same application, need a secured storage facility for aggregate data in the cloud. The data produced by each user node must be known only to itself to ensure data privacy. This in turn means that this private data must be invisible to the aggregator in the cloud server which makes the problem more challenging. However, data aggregation and data privacy protection contradict each other since to achieve data aggregation, any aggregator must view each data item they process in plaintext. End-to-end data encryption, a well-known security method, as well as popular methods of steganography has been extensively applied in this context. We propose a new method that specifically addresses such contradiction without the usage of encryption process, thereby improving the overall performance.

### *A Assumptions and Design Goals*

There is a group of user nodes that are working for the same application and send data that would be aggregated in the cloud server. We assume that each user node datum is an integer ranging from 0 to $d_{max}$. This assumption is reasonable because even if user node data are not integers in their original forms, they can still be transformed into integers.

### B Attacks Considered

Since there is always a variety of attacks to the data stored in the cloud server, there is no one magical solution for all. Our study aims at attacks that target the data transmission between user nodes and the cloud server and hence addresses data privacy preservation. To obtain the private user node data the attackers may launch a variety of attacks like

1. **Eavesdropping:** The attacker may passively eavesdrop on the message transmissions in the network.
2. **User Node Compromise:** The attacker may get control of the user node and read the stored data.
3. **User Node Colluding:** The attacker may get control of a group of user nodes and try to combine the information obtained from all of them.

### C Design Goals

The new distributed private aggregation scheme aims to achieve the following:

1. **Data privacy:** The user data collected by the user node must only be known to itself.
2. **Accuracy:** The user node data must have accurate aggregation results.
3. **Dynamic:** The proposed scheme must be adaptable to dynamic addition and deletion of user nodes since the number of user nodes participating in the node is dynamic.

The paper is organized as follows : Section II discusses the various data hiding techniques based on encryption methods, steganography and hybrid methods applied for cloud data security. We discuss in detail the proposed Enhanced Privacy preserving gene based data Aggregation Scheme(EPAS) in Section III. We conclude in Section IV.

## II RELATED WORK

In this section, the recent works contributed to securing cloud data storage over the past decades are presented with the merits and limitations. We present few works that utilize Encryption methods, steganography techniques and also hybrid methods.Singla & Singh[12] deals with the methods of providing security using data encryption and ensuring that an unauthorized intruder cannot access user file or data in the cloud. Data is encrypted by a symmetric block cipher cryptography algorithm called "Rijndael" before being stored in a cloud environment. Neha and Ganesan[14] have proposed a strategy that ensures secured transmission of data from the user to cloud server. They use Elliptic curve cryptography for data encryption and Diffie Hellman Key Exchange mechanism for connection establishment. The proposed encryption mechanism uses the combination of linear and elliptical cryptography methods. They claim that usage of ECC reduces the computational cost compared to the linear algorithms.

Kaur and Bhardwaj[1] propose a hybrid technique which combines multiple encryption algorithms such as RSA, 3DES, and random number generator for more flexibility and enhanced security in cloud data. This method suffers heavy computational time due to the multiple algorithms involved. Manivannan and Sujarani[5] have proposed a lightweight mechanism for database encryption known as transposition, substitution, folding, and shifting (TSFS) algorithm. The algorithm encrypts only the sensitive data in the database which in turn results in efficient query execution reducing user response time. Even though TSFS is designed as a symmetric encryption technique, the authors use three keys for encryption and decryption. The 3 keys used are expanded into 12 sub keys by using a key expansion technique. This method suffers from computational overhead as number of keys are increased.

Handa et.al[6] proposed an extended approach for LSB Steganography. The user selects the data to be uploaded and this selected data gets encrypted using a strong algorithm such as AES. This encrypted data is then uploaded to the cloud server. On receiving the encrypted data, a hiding algorithm is applied such that it randomly selects the bits positions from images where data is to be stored. The bit position is either 0th , 1st or 2nd position. This hiding algorithm is used to save the files or data behind the images. This process is called steganography using images. Suneetha and Kiran Kumar[13] proposed a modified Least Significant Bit (LSB) - image based Steganography approach for secure storage of information at Cloud Service Provider side. Unlike LSB steganography where only LSB is considered for storage and retrieval of data, both pixels LSB and MSB are used for storage and retrieval. Based on one of the possible values of LSB and MSB (i.e. 00, 01, 10, 11 ) bitwise or, xor operators are used for this process. This approach is tested using the metrics Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR). Rao[2] proposed a hybrid technique for data confidentiality and integrity that uses key sharing and authentication techniques. The connectivity between the user and the cloud service provider is made more secure by utilizing key sharing and authentication processes. RSA public key algorithm is used for secure distribution of the keysbetween the user and cloud service providers. Mohamed et.al.[7] proposed a three-layered data security technique in which the first layer is used for authenticity of the cloud user either by one factoror by two factor authentication. The second layer then encrypts the user data for ensuring data privacy, whereas the third layer does fast recovery of data through a decryption process. Surbhi[11] proposed a hybrid technique that utilizes encryption algorithm and steganography for data hiding in cloud. Firstly,the sensitive data is encrypted using Advanced Encryption Standard(AES) algorithm. Further in steganography,cover media is processed to form monogram puzzle inside the symmetric shapes. The selection of monogram puzzle and symmetric shape is done on the fly based on output of random number generator. Lastly,the encrypted information is hidden inside the monogram puzzle by using 2-bit LSB technique. Murat Yesilyurt and Yildiray Yalman proposed a data hiding scheme based on watermarking in order to secure the user access by the process of hiding users information by defining a coverage file that results in a highly secure covered audit [18]. This data hiding scheme was considered capable of securing the services by preventing external attacks through the encryption process enabled by the cloud architecture and deployment models used for security.

This data hiding scheme was proved to be superior in accessing security and robust during the process of storing data and access them in the time of usage. A reversible data Hiding Technique was proposed for hiding the data by applying the process of multiple encryption [19]. This reversible data hidingscheme confirmed superior recovery of hidden data by utilizing the method of histogram shuffling process. This reversible data hiding scheme was also proved to recover hidden data andcover images without any kind of errors that are possible under data exchange in cloud computing processes.

The computation in during the process of data hiding was considered to be phenomenally minimized than most of the works of the literature.

## III ENHANCED PRIVACY PRESERVING GENE BASED DATA AGGREGATION SCHEME (EPAS)

We propose a new distributed private aggregation scheme based on the constructed Enhanced P- gene for secured storage of data in cloud. The scheme adopts data hiding to achieve private data aggregation.The main contributions of the scheme are as follows:

1. We construct the Enhanced P-Gene, a data-hiding coding scheme and then propose the erasable data hiding technique. Here each user node can hide its data independently using some simple calculation operations. Then, the hidden data is sent to cloud without encrypting it, and is aggregated in the cloud server. The aggregator that implements the aggregating operation can also erase all the P-Genes without knowing them.

2. We propose a method for secret Enhanced P-Gene generation independent of cryptographic algorithms. In this method, each user node independently and dynamically generates its Enhanced P-Gene according to the dynamic reporting group members via some simple calculation operations. That is, during the generation of Enhanced P-Genes, no extra message exchange is introduced even when some or all the user nodes are reporting.

3. We focus on additive aggregation function because many other aggregation functions, including average, count, variance, standard deviation` and any other moment of the measured data, can be reduced to this function.

4. Compared the proposed work with existing data hiding approaches in cloud.

### *A Enhanced P-Gene (Privacy Preserving Gene)*

We use a group of user nodes along with the cloud server $G_a$ with size *n(which includes a cloud server and n−1 users)* to elaborate our method, in which each user node and the cloud server has a unique group ID that is selected from {1,..,*n*}. Not all user nodes may have data to report in each session. $G_a^l$ denotes the set of reporting user nodes along with the cloud server with size $m(m\leq n)$ in $G_a$ . To achieve private data aggregation, an Enhanced P-Gene is constructed for data hiding. We discuss below some concepts and properties related to the Enhanced P-Gene.

### *B P-list and P-seed*

P-list refers to the list of integers that are generated by any user node $b(b\in G_a^l)$ for enhanced P-Gene generation. These integers are denoted as P-list $\{P_c^b, c\in G_a^l\}$ and satisfies

$$\left(\sum_{c\in G_a^l} P_c^b\right) modU=0 \qquad (1)$$

where $U\geq d_{max}\times n$, $d_{max}$ being the upper bound of user data, *n* being the maximum group size that includes the cloud server and *l* being the number of bits in *U*. Each $P_c^b, c\in G_a^l$ is called P-seed and is only shared between user nodes *b* and *c*.

### *C Enhanced P-Gene*

The enhanced P-Gene of a random node $b(b\in G_a^l)$ is denoted as $R^b$

$$R^b = \left(\sum_{c\in G_a^l} P_b^c\right) modU \qquad (2)$$

**Property 1:**

Let $d^b$ denote data in user node *b*, $D^b$ denote the hidden data

$$D^b = \left(d^b + R^b\right) modU$$

For any group $G_a^l$,

$$\left(\sum_{b\in G_a^l} R^b\right) modU=0 \qquad (3)$$

$$if\left(\sum_{b\in G_a^l} d^b\right)\leq(U-1)\ then$$

$$\left(\sum_{b\in G_a^l}(d^b+R^b)\right) \mathrm{mod}\quad U = \sum_{b\in G_a^l} d^b \quad (4)$$

Property 1 shows that for any random $G_a^l$, the sum of all hidden data $D^b(b\in G_a^l)$ is equivalent to that of all original data $d^b$ under the modular addition operation.

### *D EPAS (Enhanced Privacy Preserving Gene Based Aggregation Scheme)*

We propose the enhanced privacy preserving Gene based Aggregation Scheme, that ensures data privacy in the additive data aggregation process based on the erasable data hiding technique.

### *1 Basics of Erasable data hiding Technique*

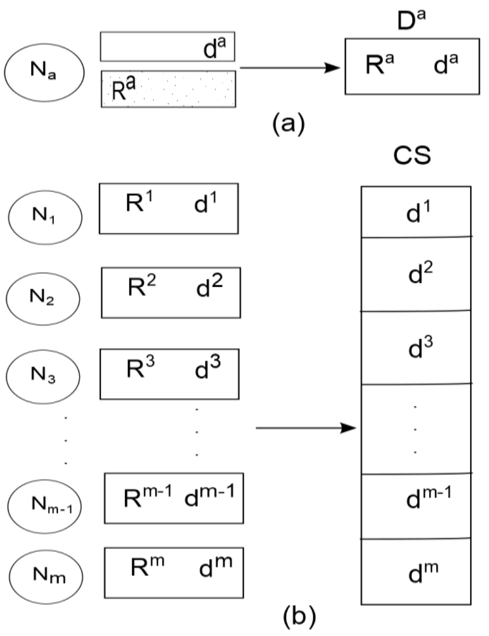


**Figure 1: EPAS Scheme**

$$\left(\sum_{a\in G_a^l}(d^a+R^a)\right) modU= \sum_{a\in G_a^l} d^a \qquad (5)$$

Fig1(a) shows how each user node hides its data using the enhanced P-Gene and sends the hidden data to the cloud server(CS). This ensures data privacy during the communication process. The cloud server receives the data from the users in

the group $G_a^l$, erases the P-Genes from the hidden data and obtains the aggregation result as shown in Fig1(b).

## *2 Numerical Example*

For illustration, let $U$=12626 and $G_a^l$={1,2,3} . The Table 1 shows the private data and P-Seeds of nodes {1,2,3}.

**Table 1: Private data and P-seeds of nodes.**

| Node b | P-Seeds | db | $R^b$ | $D^b$ |
|---|---|---|---|---|
| Node 1 | {***3654***,2319,6653} | 110 | 10750 | 10860 |
| Node 2 | {***2379***,5114,5133} | 69 | 11500 | 11569 |
| Node 3 | {***4717***,4067,3842} | 178 | 3002 | 3180 |

**Consider node1:** ( P-Genes related to Node 1 are highlighted in P-Seed Column).

According to the P-Seeds in the Table 1: Node1 calculates its PGene as follows.

$$R^b= \left(\sum_{c\in G_a^l} P_a^c\right) modU \qquad (6)$$

$R^1$=(3654+2379+4717) *mod* 12626=10750

$R^2$=(2319+5114+4067) *mod* 12626=11500

$R^3$=(6653+5133+3842) *mod* 12626=3002

for any group $G_a^l$,

$$\left(\sum_{c\in G_a^l} R^b\right) modU=0 \qquad (7)$$

There after,

$(R^1+R^2+R^3)$ *mod* 12626=0 satisfies the equation 3.

$D^b=(d^b+R^b)$ *modU*

$D^1$=(110+10750) *mod* 12626=10860

$D^2$=(69+11500) *mod* 12626=11569

$D^3$=(178+3002) *mod* 12626=3180

if

$$\left(\sum_{a\in G_a^l}(d^b)\right)\le (U-1) \qquad (8)$$

then,

$$\left(\sum_{a\in G_a^l}(d^a+R^a)\right) modU= \sum_{a\in G_a^l} d^b$$

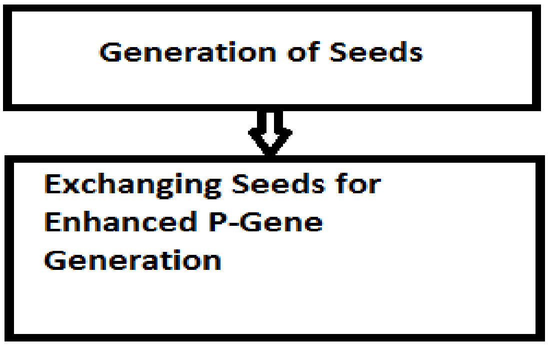
$(D^1+D^2+D^3)$ *mod* 12626=$(d^1+d^2+d^3)$

(10860+11569+3180) *mod* 12626=357

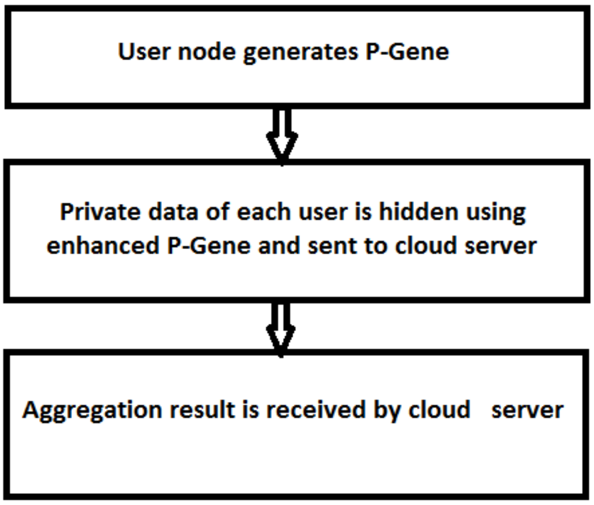
25609 *mod* 12626=357

357=357

The process of EPAS is shown in Fig 2(a) and 2 (b)



(a) Initialization Process


User node generates P-Gene
Private data of each user is hidden using enhanced P-Gene and sent to cloud server
Aggregation result is received by cloud server

(b) Data Reporting Process

**Figure 2: Process of EPAS**

Table 2 presents the basic notations used in the scheme.

**Table 2: Basic notations used in EPAS.**

| Notation | Meaning |
|---|---|
| $G_a$ | Group of cloud server and $(n-1)$user nodes |
| $G_a^l$ | Group of cloud server and $m$ participating user nodes $m\leq(n-1)$ |
| $P_c^b$ | Secret P-Seed for P-Gene generation that is generated by user $b$ and shared between user nodes $b$ and $c$. |
| $r_c^b$ | Secret seed for P-Seed generation that is generated by user $b$ and shared between user nodes $b$ and $c$. |
| $d^b$ | Original data in user node $b$. |
| $R^b$ | Enhanced P-Gene in user node $b$ |
| $D^b$ | Hidden data in user node $b$ where $D^b=(d^b+R^b)\ modU$ |

## *E Seed Table Initialization and maintenance process*

Every user node $b$, $b\in G_a$ is pre-loaded with its ID $b$ and two polynomial functions $T(x)$ and $W(x)$. The functions $T(x)$ and $W(x)$ satisfy the condition that the range of the co-efficients are $[0,U^l)$, where $U^l$ is a prime number with length $(l+L)$ bits. $T(x)$ is used in every user node $b$ to generate the secret seeds $(r_c^b)$ that is in-turn used to generate the P-Seeds $P_c^b$. $W(x)$ is used to update the seeds in case of user node deletion or addition.

The seed table is first initialized when the group of user nodes working on the same application try to store the data in the cloud server. In this case, the process is performed only once as follows:

1. **Seed Exchange:** Each user node, including the cloud server, randomly generates $(n-1)$ data as seeds $\{r_c^b,c\neq b,c\in G_a\}$ where each $r_c^b<U^l$.

Note: $U^l$ has $(l+L)$ bits and hence each $r_c^b$ has maximum $(l+L)$ bits. The user then encrypts the generated $r_c^b$ and sends to the $(n-1)$ group members through shared pairwise key

$$K_{b,c}:\{r_c^b\}K_{b,c}$$

Similarly,

node $b$ receives seeds $r_b^c$ for all $(c\neq b,c\in G_a)$ .

2. **Formation of Seeds Table:** After the initial seed exchange as done in Step 1, each node $b$ initializes its seed table $T^b$ that contains all the seeds generated by user $br_c^b\forall(b\neq c,c\in G_a)$ and all the seeds received from other group members $r_b^c\forall(b\neq c,c\in G_a)$ shown in the Table 3.

**Table 3: Seed Table $T^b$ of user node $b$.**

| c | 1 | 2 | ... | n-1 | cloud server n |
|---|---|---|---|---|---|
| $r_c^b$ | $r_1^b$ | $r_2^b$ | ... | $r_{n-1}^b$ | $r_n^b$ |
| $r_b^c$ | $r_b^1$ | $r_b^2$ | ... | $r_b^{n-1}$ | $r_b^n$ |

Updation of these seed tables is required in any of the following cases:

a. **User node failure/ compromise / deletion:** When any user node $x$ fails or has been compromised or is deleted from the group, then all other nodes $b$ in the group $G_a(b\neq x,b\in G_a)$ needs to delete the entries $r_x^b$ and $r_b^x$ from the seed table $T^b$. Also the P-seed $R^b$ needs to be updated.

b. **New user nodes have been added:** When new user nodes are added to the group $G_a$, then for each newly added user node $d$, all other user nodes $b{:}(b\in G_a)$ should generate seed $r_d^b$, add it to the seed table $T^b$ and also send to $d{:}\{r_d^b\}_{K_{b,d}}$. Similarly, every newly added user node $d$ also generates seeds for all user nodes $b$, $b\in G_a$ and sends to $b$ the encrypted seed. Each newly added user node also maintains its seed table $T^b$. Hence this process involves updation of seed table $T^b$ for all $b\in G_a$, and also generation of seed table $T^b$ for every newly added user node. Note that this updation asks for updation of P-Genes of the user nodes $b$, $b\in G_a$ and the newly added user nodes.

### *F Data Reporting Process*

In each data collection session, the user node collects data, hides its data using its P-Gene and sends the hidden data to the cloud server. The cloud server retrieves the aggregation result after receiving data from all the participating users in the group $G_a^l$. We assume that for any data collection session, the number of members in the group $G_a^l$ is $m\geq 3$.

If $m<3$, then the user node can split the data and send it to other neighboring user nodes, thereby increasing the number of participants in the session and thereby making $m>3$.

**Step 1) Original Data Hiding**

In any user node $b$, the seed table $T^b$ is available. User node $b$ generates all the P-Seeds $P_c^b\forall(c\in G_a^l, c\neq b)$, where $P_c^b$ is given by the lowest $l$ bits of $T(r_c^b)$.

Note that each $r_c^b$ is of maximum length $(l+L)$. Then the value of $P_b^b$ is computed as follows:

$$P_b^b = U - \left(\sum_{c\in G_a^l, c\neq b} P_{c,}^{b}\right) mod U$$

Also $T^b$ has entries of seeds $r_b^c\forall(c\neq b, c\in G_a^l)$ from which P-Seeds $P_b^c$ is obtained.

**Note:** $P_b^c$ is the least $l$ bits of $r_b^c$.

$b$ generates its enhanced P-Gene as follows:

$$R^b = \left(\sum_{c\in G_a^l} P_b^c\right) mod U$$

User node hides its original data $d^b$ with $R^b$ as

$$D^b = (d^b + R^b)\ mod U$$

user node $b$ then sends $(D^b, b)$ to the cloud server.

**Step 2) Data Aggregation at Cloud Server**

The cloud server checks if all user nodes of the group $G_a^l$ have sent their data. If so, it calculates

$$D = \left(\sum_{b\in G_a^l,\ b\neq CS} D^b\right) modU$$

Note that $D$ is equivalent to

$$\sum_{b\in G_a^l\ b\neq CS} d^b \qquad (9)$$

We present a simple scenario to illustrate the seed table initialization, data hiding and reporting. Given S = 2;U = 31(l = 5); $U^l$= 1021; T(x) = 179x$^2$+ 839x and $G^l_a$= {1, 2,3}. Table 4 shows the original data held by nodes 1,2,3 and their corresponding seeds.

**Table 4: Data and Seed Table $T^b$ of user nodes.**

| Node | Generated Seeds | Received Seeds | Original data |
|---|---|---|---|
| Node1 | $\{r_2^1=12, r_3^1=3\}$ | $\{r_1^2=7, r_1^3=23\}$ | $d^1=6$ |
| Node2 | $\{r_1^2=7, r_3^2=398\}$ | $\{r_2^1=12, r_2^3=756\}$ | $d^2=9$ |
| Node3 | $\{r_1^3=23, r_2^3=756\}$ | $\{r_3^1=3, r_3^2=398\}$ | $d^3=2$ |

According to $\{r_2^1=12, r_3^1=3\}, \{r_1^2=7, r_1^3=23\}$ , node 1 calculates

$T(r_2^1)\ modU^1 = (179\times 12^2 + 839\times 12)\ mod\ 1021$

$= 109 = (11\mathbf{01101})_2$

$T(r_3^1)\ modU^1 = (179\times 3^2 + 839\times 3)\ mod\ 1021$

$= 44 = (1\mathbf{01100})_2$

$T(r_1^2)\ modU^1 = (179\times 7^2 + 839\times 7)\ mod\ 1021$

$= 350 = (1010\mathbf{11110})_2$

$T(r_1^3)\ modU^1 = (179\times 23^2 + 839\times 23)\ mod\ 1021$

$= 657 = (1010\mathbf{10001})_2$

and then obtains the following

$P_2^1 = (01101)_2 = 13, \quad P_3^1 = (01100)_2 = 12,$

$P_1^2=(11110)_2=30, \quad P_1^3=(10001)_2=17$

Therefore

$$P_b^b= U-\left(\sum_{c\in G_a^l\ c\neq b} P_c^b\right) modU$$

$P_1^1=31-(P_2^1+P_3^1)\ mod\ 31=31-(13+12)\ mod\ 31=6$

$$R^b=\left(\sum_{c\in G_a^l} P_b^c\right) modU$$

$R^1=(P_1^1+P_1^2+P_1^3)\ modU=(6+30+17)\ mod\ 31$

$R^1=22$

Node 1 obtains $D^b=(d^b+R^b)\ modU$

$D^1=(6+22)\ mod\ 31=28$ and then sends {28,1} to the cloud server CS.

According to $\{r_1^2=7,r_3^2=398\},\{r_2^1=12,r_2^3=756\}$ , node 2 calculates

$T(r_1^2)\ modU^l=(179\times7^2+839\times7)\ mod\ 1021$

$=350=(1010\mathbf{11110})_2$

$T(r_3^2)\ modU^l=(179\times398^2+839\times398)\ mod\ 1021$

$=180=(101\mathbf{10100})_2$

$T(r_2^1)\ modU^l=(179\times12^2+839\times12)\ mod\ 1021$

$=109=(11\mathbf{01101})_2$

$T(r_2^3)\ modU^l=(179\times756^2+839\times756)\ mod\ 1021$

$=987=(11110\mathbf{11011})_2$

and then obtains the following

$P_1^2=(11110)_2=30, P_3^2=(10100)_2=20,$

$P_2^1=(01101)_2=13, P_2^3=(11011)_2=27$

Therefore,

$$P_b^b= U-\left(\sum_{c\in G_a^l\ c\neq b} P_c^b\right) modU$$

$P_2^2=U-(P_1^2+P_3^2)\ modU$

$P_2^2=31-(30+20)\ mod\ 31=31-19=12$

and

$R^2=(P_2^1+P_2^2+P_2^3)\ modU=(13+12+27)\ mod\ 31=21$

Node 2 obtains $D^b=(d^b+R^b)\ modU$

$D^2=(9+21)\ mod\ 31=30$then sends {30,2} to cloud server CS.

According to

$\{r_1^3=23, r_2^3=756\ \},\{r_3^1=3, r_3^2=398\}$

node 3 calculates

$T(r_1^3)\ modU^1=(179\times23^2+839\times23)\ mod\ 1021$

$=657=(1010\mathbf{10001})_2$

$T(r_2^3)\ modU^1=(179\times756^2+839\times756)\ mod\ 1021$

$=987=(11110\mathbf{11011})_2$

$T(r_3^1)\ modU^1=(179\times3^2+839\times3)\ mod\ 1021$

$=44=(1\mathbf{01100})_2$

$T(r_3^2)\ modU^1=(179\times398^2+839\times398)\ mod\ 1021$

$=180=(101\mathbf{10100})_2$

and then obtains the following

$P^3{}_1 = (10001)_2 = 17; P^3{}_2= (11011)_2 = 27;$

$P^1{}_3=(01100)_2 = 12; P^2{}_3 = (10100)_2 = 20.$

Therefore,

$$P_b^b= U-\left(\sum_{c\in G_a^l\ c\neq b} P_c^b\right) modU$$

$P_3^3=U-(P_1^3+P_2^3)\ modU$

$P_3^3=31-(17+27)\ mod31=31-(44\ mod\ 31)=18$

and

$R^3=\{P_3^1+P_3^2+P_3^3\}\ modU=12+18+20\ \mathrm{mod}\ 31=19$

Node 3 obtains$D^3=\{d^3+R^3\}\ modU=2+19\ \mathrm{mod}\ \ 31=21$ then sends {21,3} to cloud server CS.

After obtaining the data from all the user nodes {1,2,3}, CS calculates

$D^b=(D^1+D^2+D^3)modU=(28+30+21)\ mod\ 31=17$

which is equivalent to $d^b=d^1+d^2+d^3=17$ and then CS stores {17,3}.

## IV CONCLUSION

Most of the proposed works for data hiding in cloud rely on the standard encryption techniques and steganography algorithms. These algorithms have an overload of key distribution and secret key sharing. Also, the computational time taken for hiding the private data is considerably high due to the computations of the algorithms used. Hence these algorithms are not applicable to applications that require data hiding techniques with a faster computational time. A new method that specifically addresses such contradiction without the usage of encryption process is required to reduce the overheads of encryption techniques, thereby improving the overall performance. In this regard, an Enhanced Privacy preserving gene based data Aggregation Scheme (EPAS) is proposed for securing and exchanging private data by utilizing Enhanced P-Gene erasable data hiding approach. In EPAS, the use of P-Gene in data security is responsible for reducing overhead under data sharing for ensuring reliable security of data through the hidden data technique made available to the user. Numerical examples have been given that shows the correctness of the scheme proposed. Further, as future work, we plan to compare the proposed scheme with few existing data hiding schemes in Cloud as discussed in Section 2 to further comment on the improvement in computational costs and overheads incurred.

## REFERENCES

1. Kaur and M. Bhardwaj, Hybrid encryption for cloud database security,Journal of Engineering Science Technology, vol. 2, pp. 737 -741, 2012.
2. Rao, Centralized database security in cloud, International Journal of Advanced Research in Computer and Communication Engineering, vol. 1, pp. 544-549, 2012.
3. Avizienis A, Laprie J C, Randell B and Landwehr C 2004 Basic concepts and taxonomy of dependable and secure computing. IEEE Trans. Depend. Secure Comput. 1(1): 11-3
4. Balu, A., & Kuppusamy, K. (2014). An expressive and provably secure Ciphertext-Policy Attribute-Based Encryption. Information Sciences, 276(2), 354-362.
5. D. Manivannan and R. Sujarani, Light weight and secure database encryption using tsfs algorithm, in Proceedings of the International Conference on Computing Communication and Networking Technologies (ICCCNT '10), pp. 17, IEEE, 2010.
6. Handa, Karun, and Uma Singh. "Data security in cloud computing using encryption and steganography." International Journal of Computer Science and Mobile Computing 4.5 (2015): 786-791.
7. E. M. Mohamed, H. S. Abdelkader, and S. El-Etriby, Enhanced data security model for cloud computing, in Proceedings of the 8th International Conference on Informatics and Systems (INFOS '12), pp. CC-12CC-17, IEEE, 2012.
8. Liang, X., Cao, Z., Lin, H., & Xing, D. (2009). Provably secure and efficient bounded ciphertext policy attribute based encryption. Proceedings of the 4th International Symposium on Information, Computer, and Communications Security - ASIACCS '09, 1(1), 56-87.
9. Moataz, T., & Shikfa, A. (2013). Boolean symmetric searchable encryption. Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security - ASIA CCS '13, 1(2), 12-23.
10. Sarhan, A. Y., & Carr, S. (2017). A Highly-Secure Self-Protection Data Scheme in Clouds Using Active Data Bundles and Agent-Based Secure Multi-party Computation. 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud). 2(1), 59-83.
11. Singla, Surbhi. Data Embedding Technique for Image Steganography in Cloud Computing. Diss. 2018.
12. Singla, S. et J. Singh (2013). Implementing cloud data security by encryption using rijndael algorithm. Global Journal of Computer Science and Technology Cloud and Distributed 13,19-22.
13. Suneetha, D., and R. Kiran Kumar. "A Novel Algorithm for Enhancing the Data Storage Security in Cloud through Steganography." Advances in Computational Sciences and Technology 10.9 (2017): 2737-2744.
14. Tirthani, Neha, and R. Ganesan. "Data Security in Cloud Architecture Based on Diffie Hellman and Elliptical Curve Cryptography." IACR Cryptology ePrint Archive 2014 (2014): 49.
15. Waters, B. (2011). Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization. Public Key Cryptography â€“ PKC 2011, 1(1), 53-70.
16. Yang, K., Jia, X., & Ren, K. (2013). Attribute-based fine-grained access control with efficient revocation in cloud storage systems. Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security - ASIA CCS '13, 1(2), 34-47.
17. Zhu, X., Liu, Q., & Wang, G. (2016). A Novel Verifiable and Dynamic Fuzzy Keyword Search Scheme over Encrypted Data in Cloud Computing. 2016 IEEE Trustcom/BigDataSE/ISPA, 2(1), 45-59.
18. Murat Yesilyurt & Yildiray Yalman. (2016). New approach for ensuring cloud computing security: using data hiding methods, Sadhana, 1(1), 36-47.
19. Chendulkar, N. N., & Mahajani, P. (2015). Reversible Data Hiding in Cloud Based Applications. 2015 International Conference on Computational Intelligence and Communication Networks (CICN), 1(1), 24-32.

## AUTHORS PROFILE

**A. Mallareddy**, M.Tech(Ph.D) is working as Associate Professor in Department of Information Technology at CVR College of Engineering, Hyderabad, andalso pursuing Ph.D in Computer Science and Engineering from JNTUH Hyderabad. His research interests focus on Data Structures, Cloud Security, Cryptography and Network Security.

**Dr. R. Sridevi** is a Professor and heading Computer Science andEngineering Department at JNTUH College of EngineeringHyderabad, Jawaharlal Technological University Hyderabad.She received her Ph.D in 2010. Her research interests includeData Structures, Steganography, Steganalysis, Network security and Cryptography, Computer Networks and Cloud Security. She has published more than twentyresearch papers in reputed journals and eight international andnational conferences.

**Dr. Ch GVN Prasad** , M.Tech,,Ph.D(Experience-- 20 years ; 12 years IT industry ( 8 years in National Informatics Centre, Govt. of India, as Scientist and Software Analyst in AT&T in US )and 18 years Teaching as Professor and HOD of CSE dept). He is currently working as Professor in Department of Computer Science & Engineering in Sri Indu College of Engg& Tech. Hyderabad. His research interests include Network security and Cryptography, Data mining, Cloud Security. He has published more than Fifteen research papers in reputed journals , international and national conferences.