A Privacy Framework in Cloud Computing for Healthcare Data

Mayank Kumar Kundalwal
Computer Science & Engineering
National Institute of Technology Patna
Patna-800005, Bihar, (India)
Email: mayank.cspg16@nitp.ac.in

Ashish Singh
Computer Science & Engineering
National Institute of Technology Patna
Patna-800005, Bihar, (India)
Email: ashish.cse15@nitp.ac.in

Kakali Chatterjee
Computer Science & Engineering
National Institute of Technology Patna
Patna-800005, Bihar, (India)
Email: kakali@nitp.ac.in

***Abstract*—In recent years, the cloud computing technology improved the healthcare system. It includes on-demand services like storage, computing power, servers, networking, applications and many other IT resources over the internet. With the advancement of healthcare technology, the transformation from offline computing to online computing uprises many security and privacy issues such as authentication, authorization, inference control, access control, data confidentiality, abuse use of cloud based healthcare data, and integrity. As a consequence, a cloud-based e-healthcare system requires a robust and effective security and privacy framework. So, in this paper, we proposed a privacy framework which maintains the data privacy and reliability. It also ensures that only authorized party can access their corresponding healthcare data. Moreover, it overcomes the privacy issues as well as maintains a secure healthcare system. The proposed privacy framework includes a data perturbation technique, which protects the sensitive data from inference attacks. Additionally, it also includes some rule set for data view control which protects the data from impersonation attack. The detail discussion and analysis of the privacy framework shows that it is secure and efficient for healthcare data stored in the cloud.**

***Keywords*—Cloud computing, E-healthcare, Database privacy, Inference control, Data view control**

I. Introduction

In the present environment, the cloud computing can be seen as an on-demand, less effort and affordable technology which contains several physical and virtual heterogeneous entities. The goal of cloud computing is to provide sharable, scalable and cost-effective services according to user demands. With such benefits, the cloud computing technology consists of three service models, four types of cloud deployment models, and five essential characteristics [1]. The cloud computing benefits and strategic business environment are provided by many organizations such as Amazon Web Services (AWS), Google Cloud, Microsoft Azure, Digital Ocean, Limestone Networks and many more.

Due to innovative and cost-effective technology, the healthcare industries are attracted towards the benefit of cloud computing [2]. They store their sensitive data in the form of Electronic Health Records (EHR) which needs to be modified or updated very frequently. The patient's data is to be available whenever it is required. Regardless of its advantages, various organizations are worried about the adoption of cloud computing services due to security and privacy concerns [1, 3–6]. Healthcare data is very sensitive and confidential. Hence, it should be secure and only available to authorized personals only. To protect the data, several security and privacy measures such as strong authentication, query set-size restriction, inference control techniques, anonymization techniques and many more have been proposed [7–10]. But, due to different security requirements, sensitive nature of data, and less control over remote storage data, the existing security measures are inadequate for the e-healthcare systems [11]. When the healthcare organization publishes its data over the cloud, they remove all the identifiers such as name, address, SSN etc. from it to maintain the individual's privacy. But, only removing all the identifiers from the data does not ensure complete data privacy. Database Management Systems (DBMS) provides a certain level of security and protection for the stored data. But, DBMS alone cannot guarantee individual's privacy [12]. It only provides access control techniques to secure the data. The individual's privacy can be breached by performing impersonation and inference attacks on the cloud-based healthcare data. In such attacks, the adversary pretends to be a legitimate user and infer some sensitive information beyond its privileges by passing multiple authorized queries.

To provide the data privacy on digitally stored healthcare information from the above-mentioned attacks, several security measures have been introduced [10, 13–18]. The data perturbation technique and user view control by using proposed rule set is the appropriate approach to maintain the data privacy in healthcare cloud. In 2003, Kargupta et al. [19] proposed a privacy preservation technique in which random data perturbation techniques are used to secure the data from inference attack. But, under certain conditions, this approach fails to maintain the complete privacy against inference attack. In 2005, Liu et al. [18] considered the problem of computing statistical aggregates for sensitive data privacy in distributed environment. They used a multiplicative random projection matrix for distributed data privacy. They proposed an approximate random projection-based technique for privacy preserving data mining. But, this approach requires high computing power as it uses a complex mathematical equation to ensure the data privacy. In 2007, Wang et al. [20] considered matrix factorization approach to address the privacy issues in data

ISBN: 978-1-5386-4119-4/18/$31.00 ©2018 IEEE

mining techniques. The proposed efficient and flexible techniques for privacy preservation in centralized dataset provides a satisfactory performance. In 2011, Jain and Bhandare [15] analyzed the data privacy protection requirement in the data mining approach. After analyzing, they found that perturbation is the best privacy protection technique to avoid the privacy leakage of data. Hence, they proposed min-max normalization based perturbation approach to randomize the data. In 2015, Turkanovic et al.[12] discussed several types of inference attacks and their preventive measures to maintain the privacy in the statistical databases. In this paper, investigated approach shows that alone access control techniques are not sufficient to protect the data from indirect access. In 2016, Basso et al. [21] used a data perturbation model to reduce the possibility of inference and linking attacks by adding random noise to the physical data. In 2017, Mamum and Rana [13] proposed a robust authentication model using cryptographic techniques to overcome the prevailing authentication issues related to healthcare data. In this paper, the highest priority is assigned to patients to control their healthcare data.

From the above discussed existing work, we have found two major privacy issues (inference problem and user view control), which make healthcare data more vulnerable. Thus, to address the mentioned privacy issues, we have proposed a privacy framework to maintain the privacy of healthcare cloud. The proposed framework consists of the following privacy functionality.

- Randomize the query result by using hybrid data perturbation approach so that any individual's privacy cannot be breached. This technique will restrict the inference attack in the cloud database.
- A new rule-set has been proposed to control the full view of healthcare data for a legitimate user. This rule-set resists the impersonation attack on the cloud database.

The rest of the paper is structured as follows: Section II demonstrates the background of the work. Section III presents the detailed demonstration of our proposed work. Section IV describes the implementation and result part. Finally, the conclusion of the paper is described in Section V.

II. Background

In this section, we have discussed the basic concept of data perturbation which is used in our proposed framework to maintain the individual's privacy.

Data Perturbation

Data Perturbation [22, 23] is simply a technique which is generally used to maintain privacy of EHR's from inference attacks. This approach includes adding noise to the released query result. It provides results to all the passing queries without restricting any one. But, all the results are approximate. There are several data perturbation techniques such as tree based perturbation, random projection based perturbation, micro-aggregation perturbation, data output, swapping & rounding perturbation, orthogonal transformation based perturbation and many more.

For better understanding, we have presented a simplified example of data perturbation by passing a statistical query. The query to be passed is given below:

$Query$: SELECT disease, count(disease) as Query_result_without_perturbation FROM dataset GROUP BY disease;

The above query result is shown in the Table I with perturbation and without perturbation. The original query result is shown in Table I (column 2). The query result with perturbation is shown in Table I (column 3), in which some random noise is added. The additive error is computed as the difference between query result with perturbation and query result without perturbation.

TABLE I: $Query$ Result

Disease	**Query result without perturbation**	**Query result with perturbation**	**Additive error**
Asthma	378	382	4
Dengue	379	386	7
Diabetes	401	406	5
Smallpox	390	397	7
Tuberculosis	387	393	6

III. Proposed work

In this section, we have explored our proposed privacy preservation framework in detail. This framework consists of two different promising concepts- Data Perturbation and Rule-set for the user view control. This section comprises of two parts. In the first part, we have discussed the proposed hybrid data perturbation technique. The generalized rule-set is specified in the second part. The components included in the framework are users (registered_users and unregistered_users), healthcare database storage, EHR provider (rule-set provider and query validator), data perturbation module, EHR manager. All these components are structured in Figure 1. There can be two types of users: registered users and unregistered users. Registered users have to provide their credentials to login. Unregistered users can login as guest users without any credentials. The following steps are executed when any user access the healthcare cloud.

1) First the user is authenticated by EHR provider (middle tier).
2) After successful authentication, the user can pass the query to EHR provider.
3) Query validator validates the user's query and pass it to EHR manager.
4) EHR manager fetches the data from the EHR database according to the given query and transmit it to the data perturbation module.
5) Data perturbation module randomizes the fetched results according to Algorithm 1.
6) The query result will be displayed to the user according to the rule-set provided by the rule-set manager.

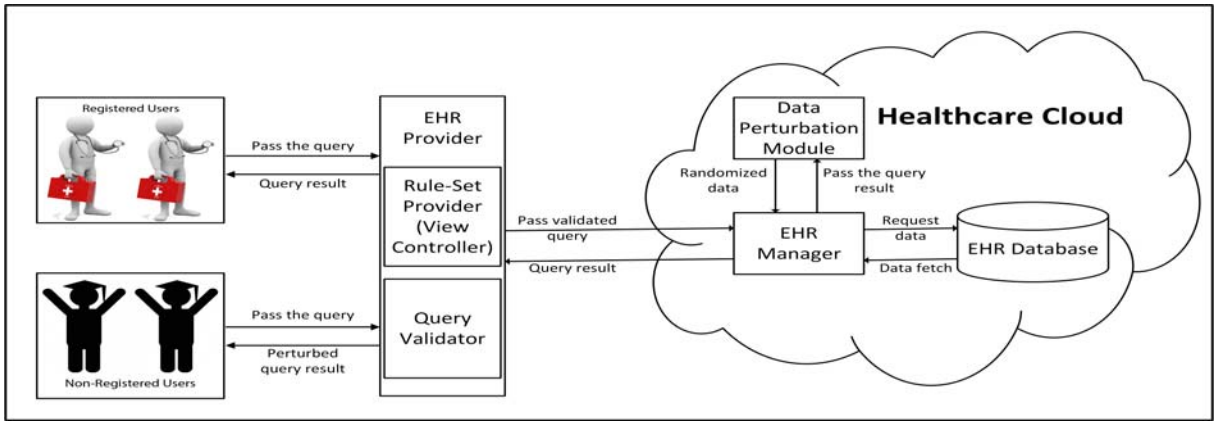

Fig. 1: Proposed privacy framework for health-cloud

Algorithm 1 Data_Perturbation

```
procedure INPUT:QUERY(Query_1)
    OUTPUT: {Query_result_with_perturbation (number of rows)}
    BEGIN
    Pass the query.
    if (query_index (Q_i) % 2 == 0) then
        query_result (Q_r) = original_data (O_d) + ((random(0,∞) % query_index (Q_i)) % 10)
    else
        query_result (Q_r) = orthonormal_matrix (O_m) * original_data (O_d)
    END
```

Proposed hybrid data perturbation technique

The fundamental aim of the proposed approach is to randomize the result output using two different data perturbation techniques- additive perturbation and multiplicative perturbation. The working of data randomization is shown in Algorithm 1. In this process, if the query index (query number) is odd, then the randomized query result is equal to the product of an orthonormal matrix and original result matrix. The mathematical expression for the randomized query result for odd number of queries is expressed by Equation 1.

$$query\_result\ (Q_r) = O_m * O_d \tag{1}$$

where O_m is an orthonormal_matrix and O_d is original_data matrix.

Orthonormal matrix [24] : The matrix $O_{m(r*r)}$ will be an orthonormal matrix, if it has the following properties.

- Let O_m^T shows the transpose of O_m, o_{ij} represent the (i, j) element of O_m, and I be the identity matrix.
- The rows and columns of matrix M are also orthonormal, i.e., for any column j, $\sum_{i=1}^{r} o_{ij}^2 = 1$
- For any two columns j and k, where j $\neq$ k then, $\sum_{i=1}^{r} o_{ij} o_{ik} = 0$.
- Similar properties will work for rows.

The Equation 2 given below can be inferred from the properties of orthonormal matrix.

$$O_m^T O_m = O_m O_m^T = I \tag{2}$$

where I is the identity matrix.

If query index (query number) is even then the query result is equal to the sum of original query result and a random number (R) generated by the Equation 3. The randomized query result will be displayed according to the Equation 4.

$$R = (random(0, \infty)/query\_index(Q_i))\%10 \tag{3}$$

$$query\_result(Q_r) = original\_data(O_d) + R \tag{4}$$

Generation of rule-set for view control

To protect the healthcare cloud data from impersonation attack, we have proposed a generalized rule-set. It also validates the context of the user before providing the results to them. The following rule-set has been introduced to maintain the privacy of healthcare cloud.

- $Rule_1$: User.Type = {Registered_User} $\bigwedge$ Query_result = {Any} $\bigwedge$ Resource = {Medical_record} $\bigwedge$ Operation = {Read, Append} $\bigwedge$ User_Location = {Stored_Location} $\bigwedge$ Time_frame = {true} $\rightarrow$ Permission = {Granted} $\bigcup$ User_view = {Non_perturbed}

Rule_Description: If any registered user wants to execute any query to read or append healthcare data from its stored location and within allowed time frame, then he/she will be allowed to view the original data.

- $Rule_2$: User.Type = {Registered_User} $\bigwedge$ Query_result = {Any} $\bigwedge$ Resource = {Medical_record} $\bigwedge$ Operation = {Read, Append} $\bigwedge$ User_Location

= {Stored_Location} $\bigwedge$ Time_frame $\neq$ {true} $\rightarrow$ Permission = {Granted} $\bigcup$ User_view = {Randomized}

Rule_Description: If any registered user wants to execute any query to read or append healthcare data from its stored location but beyond its allowed time frame, then he/she will be allowed to view only perturbed data.

- $Rule_3$: User.Type = {Registered_User} $\bigwedge$ Query_result = {Any} $\bigwedge$ Resource = {Medical_record} $\bigwedge$ Operation = {Read, Append} $\bigwedge$ User_Location $\neq$ {Stored_Location} $\bigwedge$ Time_frame = {true} $\rightarrow$ Permission = {Granted} $\bigcup$ User_view = {Randomized}

Rule_Description: If any registered user wants to execute any query to read or append healthcare data from location beyond its stored location and within allowed time frame, then he/she will be allowed to view only perturbed data.

- $Rule_4$: User.Type = {Non_Registered_User} $\bigwedge$ Query_result = {Any} $\bigwedge$ Resource = {Medical_record} $\bigwedge$ Operation = {Read} $\bigwedge$ User_Location = {Any} $\bigwedge$ Time_frame = {true} $\rightarrow$ Permission = {Granted} $\bigcup$ User_view = {Randomized}

Rule_Description: If any unregistered user wants to execute any query within time frame to read the healthcare data then he/she will be allowed to view only perturbed data.

- $Rule_5$: User.Type = {Non_Registered_User} $\bigwedge$ Query_result = {Any} $\bigwedge$ Resource = {Medical_record} $\bigwedge$ Operation = {Read} $\bigwedge$ User_Location = {Any} $\bigwedge$ Time_frame $\neq$ {true} $\rightarrow$ Permission = {Denied}

Rule_Description: If any unregistered user wants to execute any query to read the healthcare data from any location but beyond its time frame then he/she will not be allowed to view the data.

- $Rule_6$: User.Type = {DBA} $\bigwedge$ Query_result = {Any} $\bigwedge$ Resource = {Medical_record} $\bigwedge$ Operation = {Read, Append} $\bigwedge$ User_Location = {Stored_Location} $\bigwedge$ Time_frame = {true} $\rightarrow$ Permission = {Granted} $\bigcup$ User_view = {Non_perturbed}

Rule_Description: If the database administrator wants to execute any query to read or append healthcare data from its stored location and within time frame, then he/she will be allowed to view the original data.

- $Rule_7$: User.Type = {DBA} $\bigwedge$ Query_result = {Any} $\bigwedge$ Resource = {Medical_record} $\bigwedge$ Operation = {Read, Append} $\bigwedge$ User_Location = {Stored_Location} $\bigwedge$ Time_frame $\neq$ {true} $\rightarrow$ Permission = {Granted} $\bigcup$ User_view = {Randomized}

Rule_Description: If the database administrator wants to execute any query to read or append healthcare data from its stored location and beyond its allowed time frame, then he/she will be allowed to view only perturbed data.

- $Rule_8$: User.Type = {DBA} $\bigwedge$ Query_result = {Any} $\bigwedge$ Resource = {Medical_record} $\bigwedge$ Operation = {Read, Append} $\bigwedge$ User_Location $\neq$ {Stored_Location} $\bigwedge$ Time_frame = {true} $\rightarrow$ Permission = {Granted} $\bigcup$ User_view = {Randomized}

Rule_Description: If the database administrator wants to execute any query to read or append healthcare data from location beyond its stored location and within allowed time frame, then he/she will be allowed to view only perturbed data.

TABLE II: Simulation parameters for our experimental work

No.	Entity/Description	Value
1.	Experiment environment	Netbeans 8.1 and Amazon Cloud
2.	Operating system	Ubuntu 17.04
3.	Programming language	Java
4.	Database	MySQL
5.	Number of tables in database	4
6.	Dataset size (Number of tuples)	3687
7.	Dataset attributes	10

IV. Implementation and result analysis

In this section, we have implemented our privacy preservation framework. We have used our own dataset for implementation. This framework is simulated using Amazon Web Services (Amazon cloud) and Netbeans (Integrated development environment) using Java programming language. All the components and its value are tabulated in Table II. We have used our own dataset to validate the functionality of our proposed hybrid data perturbation technique. According to our rule-set, only a registered user with its stored location and within its time frame is able to view the complete data. Randomized data is provided to guest users and registered users with invalid context (location and time frame). We have compared this framework with other existing accomplished work in Table III.

In order to verify the hybrid data perturbation technique, we have passed multiple queries to get some sensitive information from the dataset, which is described subsequent paragraph. These queries and their result shows that randomized data is provided to the guest users to maintain the individual's privacy.

$Query_1$: SELECT gender, count (disease) as Query_result_without_perturbation FROM dataset where disease="Diabetes" GROUP BY gender;

$Query_1$ result is described in Table IV.

Result_Explanation: For the above query, the query index is 1 which is an odd number. Hence, according to proposed data perturbation algorithm, first, total nuber of rows(r) in output result is checked. Then an orthonormal matrix $O_{m(r*r)}$ is generated which is then multiplied to output matrix to get the randomized data. After this calculation, this randomized data is provided as a query result.

$$\begin{bmatrix} 0.5 & 0.867 \\ -0.867 & 0.5 \end{bmatrix} * \begin{bmatrix} 207 \\ 194 \end{bmatrix} = \begin{bmatrix} 272 \\ -83 \end{bmatrix}$$

Orthonormal matrix Original data Randomized data

$Query_2$: SELECT disease, count (disease) as Query_result_without_perturbation FROM dataset where disease="Diabetes" or disease="Dengue" GROUP BY disease;

$Query_2$ result is described in Table V.

TABLE III: Comparison of our privacy framework with the previous accomplished work

Paper	Year	Implemented environment	Attacks		Approach
			Inference attack (Nested queries)	Impersonation attack	
Kargupta et al. [19]	2003	Traditional database used in data mining application	✓	X	Randomly modified the result by adding noise and then constructs a random matrix-based spectral filtering technique
Liu et al. [18]	2003	Distributed data mining	✓	X	Explore independent component analysis for breaching the privacy and then proposed an approximation random projection based multiplicative data perturbation technique
Li et al. [17]	2006	Real world data set used in data mining	✓	X	Dataset are recursively partitions to convert it into the smaller subset to make more homogeneous by using KD-tree based data perturbation technique
Tamilsekvan et al. [26]	2007	Wireless mobile and ad-hoc network	X	✓	The non mutable entities are authenticated by the digital signature. The impersonation attack can be removed by the Ad hoc On-Demand Distance Vector (SAODV) security protocol.
Jain and Bhandare [15]	2011	Statistical database used in data mining	✓	X	The originality of the data will ruin to make randomized data by using min max normalization based perturbation approach
Ameen et al. [27]	2012	Wireless sensor networks for healthcare applications	X	✓	Centralized security technique for various attacks (Physical level, Administrative level and Technical level)
Abuelsaad et al. [25]	2015	Small data sets	✓	X	Data anonymization and data perturbation using one way hashing technique
Our work		**Cloud storage based health applications**	✓	✓	**Hybrid data perturbation technique in which the additive and multiplicative noise will be added according to the query number**

The "✓" and "X" symbol denote the specific aspect is covered and not covered respectively.

TABLE IV: $Query_1$ Result

Gender	Query result without perturbation	Query result with perturbation
Female	207	272
Male	194	-83

Result_Explanation: For the above query, the query index is 2 which is an even number. Hence, according to proposed data perturbation algorithm, first, a random number (4126) is generated using random() function. First, it gets divided by the query index (2) (4126/2=2063) and then it gets divided by 10 and returns the remainder i.e. (2063%10=3). After this calculation, the remainder is added to the original data to get the randomized data. Finally, this randomized data is provided as a query result.

TABLE V: $Query_2$ Result

Disease	Query result without perturbation	Query result with perturbation
Dengue	379	382
Diabetes	401	404

$Query_3$: SELECT yob , count(*) as Query_result_without_perturbation FROM dataset WHERE yob="1969" or yob="1985" or yob="1972" GROUP BY yob;

$Query_3$ result is described in Table VI.

Result_Explanation: For the above query, the query index is 3 which is an odd number. Hence, according to proposed data perturbation algorithm, first, total nuber of rows (r) in output result is checked. Then an orthonormal matrix $O_{m(r*r)}$ is generated which is then multiplied to output matrix to get the randomized data. After this calculation, this randomized data is provided as a query result.

$$\begin{bmatrix} 0.83 & -0.40 & 0.40 \\ 0.2 & 0.86 & 0.46 \\ 0.53 & 0.30 & -0.79 \end{bmatrix} * \begin{bmatrix} 48 \\ 47 \\ 47 \end{bmatrix} = \begin{bmatrix} 40 \\ 72 \\ 2 \end{bmatrix}$$

Orthonormal matrix Original data Randomized

TABLE VI: $Query_3$ Result

Year	Query result without perturbation	Query result with perturbation
1969	48	40
1972	47	72
1985	47	2

$Query_4$: SELECT disease, count (disease) as Query_result_without_perturbation FROM dataset WHERE disease="Tuberculosis" or disease="Smallpox" or disease="asthma" GROUP BY disease;

$Query_4$ result is described in Table VII.

Result_Explanation: For the above query, the query index is 4 which is an even number. Hence, according to proposed data perturbation algorithm, first, a random number (6828) is generated using random() function. First, it gets divided by the query index (4) (6828/4=1707) and then it gets divided by 10 and returns the remainder i.e. (1707%10=7). After this calculation, the remainder is added to the original data to get the randomized data. Finally, this randomized data is provided as a query result.

TABLE VII: $Query_4$ Result

Disease	Query result without perturbation	Query result with perturbation
Asthma	378	385
Smallpox	390	397
Tuberculosis	387	394

V. Conclusion

Data perturbation based privacy technique plays an important role to preserve individuals privacy in healthcare cloud. However, this paper illustrates a robust and secure privacy framework for the cloud-based e-healthcare system, which suffers from several attacks such as inference attacks and impersonation attacks. Additionally, we have also included a rule set for full view control which prevents the EHRs from impersonation attacks. We have also examined our simulated work by passing several nested queries to get the sensitive knowledge from the database. The experimental demonstration of our privacy preservation framework shows that attacker is not able to get the knowledge referring to some individual. Hence, this framework restricts the adversary and prevents the data from inference and impersonation attacks.

References

[1] Chirag Modi, Dhiren Patel, Bhavesh Borisaniya, Avi Patel, and Muttukrishnan Rajarajan. A survey on security issues and solutions at different layers of cloud computing. *The journal of supercomputing*, 63(2):561–592, 2013.

[2] Hajar Ziglari and Arefeh Negini. Evaluating cloud deployment models based on security in ehr system. In *Engineering and Technology (ICET), 2017 International Conference on*, pages 1–6. IEEE, 2017.

[3] Mahroosh Irfan, Muhammad Usman, Yan Zhuang, and Simon Fong. A critical review of security threats in cloud computing. In *Computational and Business Intelligence (ISCBI), 2015 3rd International Symposium on*, pages 105–111. IEEE, 2015.

[4] Farhan Bashir Shaikh and Sajjad Haider. Security threats in cloud computing. In *Internet technology and secured transactions (ICITST), 2011 international conference for*, pages 214–219. IEEE, 2011.

[5] Prachi Deshpande, SC Sharma, and P Sateesh Kumar. Security threats in cloud computing. In *Computing, Communication & Automation (ICCCA), 2015 International Conference on*, pages 632–636. IEEE, 2015.

[6] Hassan Takabi, James BD Joshi, and Gail-Joon Ahn. Security and privacy challenges in cloud computing environments. *IEEE Security & Privacy*, 8(6):24–31, 2010.

[7] Nabil R. Adam and John C. Worthmann. Security-control methods for statistical databases: A comparative study. *ACM Comput. Surv.*, 21(4):515–556, December 1989.

[8] Leland L. Beck. A security machanism for statistical database. *ACM Trans. Database Syst.*, 5(3):316–3338, September 1980.

[9] John B. Kam and Jeffrey D. Ullman. A model of statistical database their security. *ACM Trans. Database Syst.*, 2(1):1–10, March 1977.

[10] Mubina Malik and Trisha Patel. Database security-attacks and control methods. *International Journal of Information Sciences and Techniques (IJIST)*, 6(1/2), 2016.

[11] Repu Daman, Manish M Tripathi, and Saroj K Mishra. Security issues in cloud computing for healthcare. In *Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on*, pages 1231–1236. IEEE, 2016.

[12] Muhamed Turkanovic, Tatjana Welzer Druzovec, and Marko Hölbl. Inference attacks and control on database structures. *TEM Journal*, 4(1):3, 2015.

[13] Quazi Mamun and Muhammad Rana. A robust authentication model using multi-channel communication for ehealth systems to enhance privacy and security. In *Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2017 8th IEEE Annual*, pages 255–260. IEEE, 2017.

[14] Asaph Azaria, Ariel Ekblaw, Thiago Vieira, and Andrew Lippman. Medrec: Using blockchain for medical data access and permission management. In *Open and Big Data (OBD), International Conference on*, pages 25–30. IEEE, 2016.

[15] Y Kumar Jain and Santosh Kumar Bhandare. Min max normalization based data perturbation method for privacy protection. *International Journal of Computer & Communication Technology*, 2(8):45–50, 2011.

[16] Matthias Mettler. Blockchain technology in healthcare: The revolution starts here. In *e-Health Networking, Applications and Services (Healthcom), 2016 IEEE 18th International Conference on*, pages 1–3. IEEE, 2016.

[17] Xiao-Bai Li and Sumit Sarkar. A tree-based data perturbation approach for privacy-preserving data mining. *IEEE Transactions on Knowledge and Data Engineering*, 18(9):1278–1283, 2006.

[18] Kun Liu, Hillol Kargupta, and Jessica Ryan. Random projection-based multiplicative data perturbation for privacy preserving distributed data mining. *IEEE Transactions on knowledge and Data Engineering*, 18(1):92–106, 2006.

[19] Hillol Kargupta, Souptik Datta, Qi Wang, and Krishnamoorthy Sivakumar. On the privacy preserving properties of random data perturbation techniques. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 99–106. IEEE, 2003.

[20] Jie Wang and Jun Zhang. Addressing accuracy issues in privacy preserving data mining through matrix factorization. In *Intelligence and Security Informatics, 2007 IEEE*, pages 217–220. IEEE, 2007.

[21] Tania Basso, Roberta Matsunaga, Regina Moraes, and Nuno Antunes. Challenges on anonymity, privacy, and big data. In *Dependable Computing (LADC), 2016 Seventh Latin-American Symposium on*, pages 164–171. IEEE, 2016.

[22] Krishnamurty Muralidhar, Rahul Parsa, and Rathindra Sarathy. A general additive data perturbation method for database security. *management science*, 45(10):1399–1415, 1999.

[23] Josep Domingo-Ferrer and Vicenc Torra. Disclosure control methods and information loss for microdata. *Confidentiality, disclosure, and data access: theory and practical applications for statistical agencies*, pages 91–110, 2001.

[24] Lorenzo Adlai Sadun. *Applied linear algebra : the decoupling principle*. Providence, R.I. : American Mathematical Society ; Oxford : Oxford University Press [distributor], 2nd ed edition, 2008. Formerly CIP.

[25] Tamer E Abuelsaad and Carlos Hoyos. Data perturbation and anonymization using one way hash, December 1 2015. US Patent 9,202,078.

[26] Latha Tamilselvan and Dr V Sankaranarayanan. Prevention of impersonation attack in wireless mobile ad hoc networks. *International Journal of Computer Science and Network Security (IJCSNS)*, 7(3):118–123, 2007.

[27] Moshaddique Al Ameen, Jingwei Liu, and Kyungsup Kwak. Security and privacy issues in wireless sensor networks for healthcare applications. *Journal of medical systems*, 36(1):93–101, 2012.