International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048
Volume 8, Issue 2, pp 148-154, February-2024
https://doi.org/10.47001/IRJIET/2024.802022

Privacy Preserving Many-Sided Shield in Cloud Environment

[1]Anu.T. S., [2]Prof. P. Gopika

[1]PG Student, Dept. of Computer Science and Engineering, EASA College of Engineering and Technology, Tamilnadu, India
[2]Professor, Dept. of Computer Science and Engineering, EASA College of Engineering and Technology, Tamilnadu, India

***Abstract* - In the current cloud computing scenario keyword-based search over encrypted outsourced data has become an important tool. The majority of the existing techniques are focusing on multi-keyword exact match or single keyword fuzzy search. However, those existing techniques find less practical significance in real world applications compared with the multi-keyword fuzzy search technique over encrypted data. The first attempt to construct such a multi-keyword fuzzy search scheme was reported by Wang et al., who used locality-sensitive hashing functions and Bloom filtering to meet the goal of multi-keyword fuzzy search. Nevertheless, Wang's scheme was only effective for a one letter mistake in keyword but was not effective for other common spelling mistakes. Moreover, Wang's scheme was vulnerable to server out-of-order problems during the ranking process and did not consider the keyword weight. In this project, based on Wang et al.'s scheme, we propose an efficient multi-keyword fuzzy ranked search scheme based on Wang et al.'s scheme that is able to address the aforementioned problems. First, we develop a new method of keyword transformation based on the uni-gram, which will simultaneously improve the accuracy and creates the ability to handle other spelling mistakes. In addition, keywords with the same root can be queried using the stemming algorithm. Furthermore, we consider the keyword weight when selecting an adequate matching file set. Experiments using real-world data show that our scheme is practically efficient and achieve high accuracy.**

Keywords: Outsourcing security, privacy preserving, searchable encryption, multi-keyword search, fuzzy search.

I. INTRODUCTION

Besides, in order to improve feasibility and save on the expense in the cloud paradigm, it is preferred to get the retrieval result with the most relevant files that match users' interest instead of all the files, which indicates that the files should be ranked in the order of relevance by users' interest and only the files with the highest relevance's are sent back to users. A series of searchable symmetric encryption schemes have been proposed to enable search on cipher text. Traditional SSE schemes enable users to securely retrieve the cipher text, but these schemes support only Boolean keyword search, i.e., whether a keyword exists in a file or not, without considering the difference of relevance with the queried keyword of these files in the result. Preventing the cloud from involving in ranking and entrusting all the work to the user is a natural way to avoid information leakage. However, the limited computational power on the user side and the high computational overhead precludes information security.

Due to the flexibility and economic savings offered by the cloud server, the users have been motivated to outsource the management of their data to the cloud. However, because of privacy concerns, data owners encrypt sensitive data prior to outsourcing, which in turn makes data utilization a challenging problem. Thus, development of an efficient privacy preserving search system over encrypted cloud data is of great importance. The most common search methods retrieve files using keywords instead of retrieving all the encrypted files back. To securely searching over encrypted data, the data owner usually builds an encrypted index structure using the extracted keywords from the data files and a corresponding index-based keyword matching algorithm and subsequently outsources both the encrypted data and this constructed index structure to the cloud.

When searching the files, the cloud server integrates the trapdoors of the keywords with the index information and then returns the corresponding files to the data users. Moreover, the data owner can share their data with a large number of users which requires the cloud server to have the ability to meet a large amount of requests with effective data retrieval services. One effective method for solving this problem is ranking the results and sending back the top-K files to the data user, rather than all of the relevant files. This method can dramatically reduce the communication overhead and still meet user's demand. However, such a ranking operation should not leak any other information related to the keywords.

In recent years, many efforts have been directed toward the design of efficient mechanisms for searching over encrypted data. Many of these methods only supported single keyword search and others simply offered conjunctive or

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 2, pp 148-154, February-2024

https://doi.org/10.47001/IRJIET/2024.802022

disjunctive searches for multi-keyword queries. We note that these schemes support only exact keyword matching. As a result, if keywords are misspelled, incorrect results are returned. Only a few efforts have achieved the fuzzy keyword search.

Wang et al.'s work was one of the first works to address the problem of multi-keyword fuzzy search over encrypted data problem and did not require a predefined fuzzy set (referred to as MFSE). In MFSE, a keyword was first transformed into a bigram set and the Euclidean distance was used to capture keywords similarity. The MFSE subsequently used LSH functions from the same hash family to generate the Bloom filter based index and query. Due to the nature of LSH, even if the keyword was misspelled, it still could be hashed into the corresponding bits in the query vectors with high probability. Finally, this method used the inner product of the index vector and the query vector as the relevance score between queries and documents. This scheme solved the problems of multi-keyword fuzzy search with high efficiency and accuracy. The most important result was that their scheme does not require the predefined fuzzy set. However, some other problems arose in this scheme. First, converting the keyword into a bi-gram set will increases the Euclidean distance. For example, for the misspelled keyword "netward", the bi-gram set is {ne, et, tw,wa, ar, rd}. Two bigram sets are different compared with original sets, which means that the Euclidean distance between two vectors is 2, and the vectors are rarely be thresholded into the same bit. Second, the scheme is not effective for other spelling mistakes. In fact, a keyword can be misspelled into many forms, not only one-letter mistakes. For example, the keyword "network" can be misspelled as "network" "network" or "network". All of the above-mentioned spelling mistakes are common and should be considered by the search system. In addition, this scheme could not find the keywords with same root such as "walk" and "walking". Finally, MFSE did not show the relevance between the keywords and files. For the same keyword in different files, its keyword weight should be different, and this difference should be considered during ranking. Thus, the files that are more relevant to the query keyword might not be included in the return results. To overcome the issues listed above, we develop a new multi-keyword fuzzy ranked search scheme based on MFSE. Our contributions can be summarized as follows:

- We develop a novel method of keyword transformation based on the uni-gram. For misspelling of one letter, this method reduce the Euclidean distance between the misspelled keyword and the correct keyword. Moreover, this method is also effective for other spelling mistakes. Additionally, we introduce the stemming algorithm to obtain the root of the word. Using this technique, the keywords with the same root can also be queried.
- We take the keyword weight into consideration in constructing the ranked list of the results. The files that are more relevant to the keywords will have greater chances to appear first on the list.
- We implement and evaluate our proposed scheme using a real world data set. The results demonstrate that our proposed scheme efficiently achieves high accuracy.

II. PROBLEM FORMULATION

We formulate the privacy problem of the multi-key word fuzzy ranked search over encrypted data in this section.

A) System Model

In this paper, we consider a cloud system consisting of data owner, data user and cloud server, see Fig. 1. In our system model, data owner has a collection of n data files F = (F1, F2, F3, • • • ,Fn) and outsources them to the cloud server in the encrypted form C. To enable efficient search operation on these encrypted files, data owner will build a secure searchable index I on the keyword set W extracted from F. Both the index I and the encrypted data files C, are outsourced to the cloud server. To search the encrypted data files for t given keywords, an authorized user computes a corresponding trapdoor T and sends it to cloud server. Upon receiving the trapdoor, the cloud server is responsible to search the index I and return the corresponding set of the encrypted documents. To improve the file retrieval accuracy and save the communication cost, the search result should be ranked by the cloud server and return the top-K relevant files to the user as the search results.

B) Threat Model

In our threat model, both data owners and data users are trusted. However, the cloud server is honest-but-curious. Even though data files are encrypted, the cloud server may try to obtain other sensitive information from user search requests while performing keyword-based search over C. So the search should be performed in a secure manner that allows data files to be securely retrieved while revealing as little information as possible to the cloud. We consider the threat models as follow:

1) Know Ciphertext Model:

The cloud server can only know the encrypted files, the secure index and the submitted trapdoors. The cloud server can also know and record the search results. The semantic meaning of this threat scenario is captured by the non-adaptive attack model.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 2, pp 148-154, February-2024

https://doi.org/10.47001/IRJIET/2024.802022

2) Known Background Model:

The cloud server knows additional background information in this model. The background refers to the information which can be learned from a comparable dataset. For example, the correlation relationship of two given trapdoors. In this Model, the cloud server can use scale analysis to deduce the keyword specific information, which can be further combined with background information to identify the keyword in a query at high probability. C. Design Goals:

III. BASIC IDEA OF OUR SCHEME

We develop a novel method of keyword transformation based on the uni-gram. For misspelling of one letter, this method reduce the Euclidean distance between the misspelled keyword and the correct keyword. Moreover, this method is also effective for other spelling mistakes. Additionally, we introduce the stemming algorithm to obtain the root of the word. Using this technique, the keywords with the same root can also be queried. We take the keyword weight into consideration in constructing the ranked list of the results. The files that are more relevant to the keywords will have greater chances to appear first on the list. We implement and evaluate our proposed scheme using a real world data set. The results demonstrate that our proposed scheme efficiently achieves high accuracy.

A) Multi-Keyword Fuzzy Search

One of the first works to address the problem of multi-keyword fuzzy search over encrypted data problem and did not require a predefined fuzzy set, referred to as MFSE. In MFSE, a keyword was first transformed into a bi-gram set and the Euclidean distance was used to capture keywords similarity. The MFSE subsequently used LSH functions from the same hash family to generate the Bloom filter based index and query. Due to the nature of LSH, even if the keyword was misspelled, it still could be hashed into the corresponding bits in the query vectors with high probability. Finally, this method used the inner product of the index vector and the query vector as the relevance score between queries and documents. This scheme solved the problems of multi-keyword fuzzy search with high efficiency and accuracy. The most important result was that their scheme does not require the predefined fuzzy set.

1) Keyword Search Algorithm

In computer science, a search algorithm is an algorithm that retrieves information stored within some data structure. Data structures can include linked lists, arrays, search trees, hash tables, or various other storage methods. The appropriate search algorithm often depends on the data structure being searched. Searching also encompasses algorithms that query the data structure, such as the SQL SELECT command. Search algorithms can be classified based on their mechanism of searching. Linear search algorithms check every record for the one associated with a target key in a linear fashion. Binary, or half interval searches, repeatedly target the centre of the search structure and divide the search space in half. Comparison search algorithms improve on linear searching by successively eliminating records based on comparisons of the keys until the target record is found, and can be applied on data structures with a defined order. Digital search algorithms work based on the properties of digits in data structures that use numerical keys. Finally, hashing directly maps keys to records based on a hash function. Searches outside of a linear search require that the data be sorted in some way.

Search functions are also evaluated on the basis of their complexity, or maximum theoretical run time. Binary search functions, for example, have a maximum complexity of O(log(n)), or logarithmic time. This means that the maximum number of operations needed to find the search target is a logarithmic function of the size of the search space.

2) Permutation Search

Permutation search is a useful tool that searches for terms that contain the set of keywords ordered in different sequences. This is a more restricted and targeted search, and is particularly useful when you are trying to target a specific group of keywords. To use this feature, the keywords must be entered separated by commas. "n/a" in the search column indicates that the keyword has no searches registered in the Keyword Discovery database.

B) Scoring and Ranking

Some of the multi-keyword searchable symmetric encryption schemes support only Boolean queries, i.e., a file either matches or does not match a query. Considering the large number of data users and documents in the cloud, it is necessary to allow multi-keyword in the search query and return documents in the order of their relevancy with the queried keywords. Scoring is a natural way to weight the relevance. Based on the relevance score, files can then be ranked in either ascendingly or descendingly. Several models have been proposed to score and rank files in information retrieval (IR) community.

1) Computing Vector Score

In a typical setting we have a collection of documents each represented by a vector, a free text query represented by a vector, and a positive integer k. We seek the k documents of

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 2, pp 148-154, February-2024

https://doi.org/10.47001/IRJIET/2024.802022

the collection with the highest vector space scores on the given query. Typically, we seek these k top documents in ordered by decreasing score. The array length holds the lengths (normalization factors) for each of the N- documents, w here as the array scores holds the scores for each of the documents.

When the scores are finally computed in step 9, all that remains in step 10 is to pick off the k-documents with the highest scores. The outermost loop beginning step 3 repeats the updating of scores, iterating over each query term-t in turn. In step 5 we calculate the weight in the query vector for term-t. Step 6-8update the score of each document by adding in the contribution from term-t. This sometimes known as term-at-a-time scoring or accumulation, and the N elements of the array scores are therefore known as accumulators. For this purpose, it would appear necessary to store, with each postings entry, the weight wft, d of term-t in document-d.

2) Data Mining Algorithms

An algorithm in data mining (or machine learning) is a set of heuristics and calculations that creates a model from data. To create a model, the algorithm first analyses the data you provide, looking for specific types of patterns or trends. The algorithm uses the results of this analysis over much iteration to find the optimal parameters for creating the mining model. These parameters are then applied across the entire data set to extract actionable patterns and detailed statistics.

The mining model that an algorithm creates from your data can take various forms, including:

- A set of clusters that describe how the cases in a dataset are related.
- A decision tree that predicts an outcome, and describes how different criteria affect that outcome.
- A mathematical model that forecasts sales.
- A set of rules that describe how products are grouped together in a transaction, and the probabilities that products are purchased together.

The algorithms provided in SQL Server Data Mining are the most popular, well-researched methods of deriving patterns from data. To take one example, K-means clustering is one of the oldest clustering algorithms and is available widely in many different tools and with many different implementations and options. However, the particular implementation of K-means clustering used in SQL Server Data Mining was developed by Microsoft Research and then optimized for performance with Analysis Services. All of the Microsoft data mining algorithms can be extensively customized and are fully programmable, using the provided APIs. You can also automate the creation, training, and retraining of models by using the data mining components in Integration Services.

You can also use third-party algorithms that comply with the OLE DB for Data Mining specification, or develop custom algorithms that can be registered as services and then used within the SQL Server Data Mining framework.

3) C4.5 algorithm

C4.5 is an algorithm used to generate a decision tree developed by Ross Quinlan. C4.5 is an extension of Quinlan's earlier ID3 algorithm. The decision trees generated by C4.5 can be used for classification, and for this reason, C4.5 is often referred to as a statistical classifier.

At each node of the tree, C4.5 chooses the attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. The splitting criterion is the normalized information gain, difference in entropy. The attribute with the highest normalized information gain is chosen to make the decision. The C4.5 algorithm then recurs on the smaller subsists.

This algorithm has a few base cases.

- All the samples in the list belong to the same class. When this happens, it simply creates a leaf node for the decision tree saying to choose that class.
- None of the features provide any information gain. In this case, C4.5 creates a decision node higher up the tree using the expected value of the class.
- Hybrid Feature Selection Method [1] was proposed to select the best feature based on the iteration manner, but with the EXPECTED LIMI.
- Instance of previously-unseen class encountered. Again, C4.5 creates a decision node higher up the tree using the expected value.

D) Security Intend Computational Encryption

To alleviate the computational burden on user side, computing work should be at the server side, so we need an encryption scheme to guarantee the operability and security at the same time on server side. Homomorphic encryption allows specific types of computations to be carried out on the corresponding cipher text. The result is the cipher text of the result of the same operations performed on the plaintext. That is, homomorphic encryption allows computation of cipher text without knowing anything about the plaintext to get the correct encrypted result. Although it has such a fine property, original fully homomorphic encryption scheme, which employs ideal lattices over a polynomial ring, is too complicated and inefficient for practical utilization.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 2, pp 148-154, February-2024

https://doi.org/10.47001/IRJIET/2024.802022

Fortunately, as a result of employing the vector space model to top-k retrieval, only addition and multiplication operations over integers are needed to compute the relevance scores from the encrypted searchable index. Therefore, we can reduce the original homomorphism in a full form to a simplified form that only supports integer operations, which allows more efficiency than the full form does. 1) Encoding/Decoding Algorithms.

The Rijndael algorithm is a new generation symmetric block cipher that supports key sizes of 128, 192 and 256 bits, with data handled in 128-bit blocks - however, in excess of AES design criteria, the block sizes can mirror those of the keys. Rijndael uses a variable number of rounds, depending on key/block sizes, as follows:

9 rounds if the key/block size is 128 bits

11 rounds if the key/block size is 192 bits

13 rounds if the key/block size is 256 bits

Rijndael is a substitution linear transformation cipher, not requiring a Feistel network. It uses triple discreet invertible uniform transformations (layers). Specifically, these are: Linear Mix Transform; Nonlinear Transform and Key Addition Transform. Even before the first round, a simple key addition layer is performed, which adds to security. Thereafter, there are Nr-1 rounds and then the final round. The transformations form a State when started but before completion of the entire process.

IV. RELATED WORK

1) Bing Wang, Shucheng Yu, Wenjing Lou, Y. Thomas Hou," Privacy-Preserving Multi-Keyword Fuzzy Search over Encrypted Data in the Cloud". Enabling keyword search directly over encrypted data is a desirable technique for effective utilization of encrypted data outsourced to the cloud. Existing solutions provide multi keyword exact search that does not tolerate keyword spelling error, or single keyword fuzzy search that tolerates typos to certain extent. The current fuzzy search schemes rely on building an expanded index that covers possible keyword misspelling, which lead to significantly larger index file size and higher search complexity.

2) Mengmeng Li, Guijuan Wang, Suhui Liu, Jiguo Yu,"Multi-keyword Fuzzy Search over Encrypted Cloud Storage Data". With the rapid development of the cloud computing, more and more users choose to store their encrypted data on the cloud server. Searchable encryption schemes capacitate legal users to retrieve encrypted data in the cloud server and the cloud server does not leak sensitive information about the encrypted data. In this paper, we use searchable symmetric encryption to construct our scheme.

3) Jie Wang, Xiao Yu , Ming Zhao ,"Privacy-Preserving Ranked Multi-keyword Fuzzy Search on Cloud Encrypted Data Supporting Range Query". It is a desirable technique for cloud users to make the fullest use of cloud encrypted data by searching what they need through input keywords. Exact keyword search schemes over encrypted data have been well tackled for better retrieval efficiency and accuracy. However, existing researches on fuzzy keyword search are mainly based on single-input keyword, where multi-keyword fuzzy search remains to be unsolved, and keyword-based search application expansion. The information access pattern can be helpful in identifying the learning behavior traits of an individual.

4) Hassan El Gafif and Ahmed Toumanari," Efficient Ciphertext-Policy Attribute-Based Encryption Constructions with Outsourced Encryption and Decryption". The invention of the Ciphertext-Policy Attribute-Based Encryption scheme opened a new perspective for realizing attribute-based access control systems without being forced to trust the storage service provider, which is the case in traditional systems where data are sent to the storage service provider in clear and the storage service provider is the party that controls the access to these data.

5) Naresh Vurukonda ,Venkateshwarlu Velde, M.Trinath Basu, P.Tejasri," Simplified ciphertext-policy attribute-based encryption scheme with attribute level collusion resistance for cloud storage". There has recently been an increasing need for the collection and sharing of microdata containing information regarding an individual entity. Because microdata typically contain sensitive information on an individual, releasing it directly for public use may violate existing privacy requirements. Thus, extensive studies have been conducted on privacy-preserving data publishing, which ensures that any microdata released satisfy the privacy policy requirements.

V. CONCLUSION

In this project, we have proposed an efficient and fine-grained data access control scheme for big data, where the access policy will not leak any privacy information. Different from the existing methods which only partially hide the attribute values in the access policies, our method can hide the whole attribute (rather than only its values) in the access policies. However, this may lead to great challenges and difficulties for legal data consumers to decrypt data. To cope with this problem, we have also designed an attribute localization algorithm to evaluate whether an attribute is in the access policy. In order to improve the efficiency, a novel Attribute Bloom Filter has been designed to locate the precise

ISSN (online): 2581-3048
Volume 8, Issue 2, pp 148-154, February-2024
https://doi.org/10.47001/IRJIET/2024.802022

row numbers of attributes in the access matrix. We have also demonstrated that our scheme is selectively secure against chosen plaintext attacks. Moreover, we have implemented the abf by using Murmur Hash and the access control scheme to show that our scheme can preserve the privacy from any lsss access policy without employing much overhead. In our future work, we will focus on how to deal with the offline attribute guessing attack that check the guessing "attribute strings" by continually querying the ABF.

VI. FUTURE WORK

We consider the problem of multi keyword fuzzy ranked search over encrypted cloud data. We propose a multi-keyword fuzzy ranked search scheme based on Wang et al.'s scheme. Concretely, we develop a novel method of keyword transformation and introduce the stemming algorithm. With these two techniques, the proposed scheme is able to efficiently handle more misspelling mistake. Moreover, our proposed scheme takes the keyword weight into consideration during ranking. Like Wang et al.'s scheme, our proposed scheme does not require a predefined keyword set and hence enables efficient file update too. We also give thorough security analyses and conduct experiments on real world data set, which indicates the proposed scheme's potential of practical usage. When the user's query is a sentence, we can extract the attributes of a sentence, and then express the relationship between attributes and search though the attributes, it is called as semantic search. We failed to achieve the ideal state because of the keyword weight. We will develop a way to reflect the keyword weight and enable update. We will design a verifiable search scheme over encrypted cloud data. Nowadays, many works were mainly focusing on the cases of single data owner.

REFERENCES

[1] B. Wang, S. Yu, W. Lou, and Y. T. Hou, "Privacy-preserving multikeyword fuzzy search over encrypted data in the cloud," in Proc. IEEE INFOCOM, Apr./May 2014, pp. 2112–2120.

[2] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," in Proc. IEEE INFOCOM, Apr. 2011, pp. 829–837.

[3] W. Sun et al., "Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking," in Proc. 8th ASIACCS, 2013, pp. 71–82.

[4] Z. Xu, W. Kang, R. Li, K. C. Yow, and C.-Z. Xu, "Efficient multikeyword ranked query on encrypted data in the cloud," in Proc. 18th IEEE Int. Conf. Parallel Distrib. Syst., Dec. 2012, pp. 244–251.

[5] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A secure and dynamic multi keyword ranked search scheme over encrypted cloud data," IEEE Trans. Parallel Distrib. Syst., vol. 27, no. 2, pp. 340–352, Feb. 2016, doi: 10.1109/TPDS.2015.2401003.

[6] Z. Fu, J. Shu, X. Sun, and D. Zhang, "Semantic keyword search based on trie over encrypted cloud data," in Proc. 2nd Int. Workshop Security Cloud Comput., Kyoto, Japan, Jun. 2014, pp. 59–62.

[7] Z. Fu, J. Shu, X. Sun, and N. Linge, "Smart cloud search services: Verifiable keyword-based semantic search over encrypted cloud data," IEEE Trans. Consum. Electron., vol. 60, no. 4, pp. 762–770, Nov. 2014.

[8] M. Chuah and W. Hu, "Privacy-aware bedtree based solution for fuzzy multi-keyword search over encrypted data," in Proc. 31st Int. Conf. Distrib. Comput. Syst. Workshops (ICDCSW), Jun. 2011, pp. 273–281.

[9] C. Liu, L. Zhu, L. Li, and Y. Tan, "Fuzzy keyword search on encrypted cloud storage data with small index," in Proc. ICCCIS, Sep. 2011, pp. 269–273.

[10] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: Improved definitions and efficient constructions," in Proc. CCS, 2006, pp. 79–88.

[11] M. Kuzu, M. S. Islam, and M. Kantarcioglu, "Efficient similarity search over encrypted data," in Proc. 28th IEEE Int. Conf. Data Eng., Washington, DC, USA, Apr. 2012, pp. 1156–1167.

[12] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in Proc. IEEE INFOCOM, Mar. 2010, pp. 1–5.

[13] Dr. Anna Saro Vijendran D. Nethra Pingala Suthishni "A Novel Approach for Intrusion Detection Using DSR Protocol in Mobile Ad hoc Networks" Journal of Advanced Research in Dynamical & Control Systems (JARDCS), VOLUME: 11.

[14] DR.K.PRABAVATHY K.S.SENTHILKUMAR "An Adaptive Scheme In Under Water Sensor Network With Eavarp-Clique "INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH MARCH 2020.

[15] E. Chandra Blessie, S. Gnanapriya "Robustious Feature Selection Based Genetic Algorithm (RFS-GA) For Cross Domain Opinion MINING", International Journal of Recent Technology and Engineering (IJRTE), ISSN: 2277-3878, Volume-8, Issue-2, July 2019.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 2, pp 148-154, February-2024

https://doi.org/10.47001/IRJIET/2024.802022

Citation of this Article:

Anu.T. S., Prof. P. Gopika, "Privacy Preserving Many-Sided Shield in Cloud Environment" Published in *International Research Journal of Innovations in Engineering and Technology - IRJIET*, Volume 8, Issue 2, pp 148-154, February 2024. Article DOI https://doi.org/10.47001/IRJIET/2024.802022

****

