**Improvement of a PIN-Entry Method Resilient to Shoulder Surfing and Recording Attacks in ATM**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | **TITLE** | **PAGE NO** |
|  | **ABSTRACT** |  |
|  | **INTRODUCTION** |  |
|  | * 1. General Introduction |  |
|  | * 1. Project Objectives |  |
|  | * 1. Problem Statement |  |
|  | **SYSTEM** **PROPOSAL** |  |
|  | * 1. Existing System |  |
|  | * + 1. Disadvantages |  |
|  | * 1. Proposed System |  |
|  | 2.2.1 Advantages |  |
|  | **SYSTEM DIAGRAMS** |  |
|  | * 1. Architecture Diagram |  |
|  | * 1. Flow Diagram |  |
|  | * 1. UML Diagrams |  |
|  | **IMPLEMENTATION** |  |
|  | * 1. Modules |  |
|  | * 1. Modules Description |  |
|  | **SYSTEM** **REQUIREMENTS** |  |
|  | * 1. Hardware Requirements |  |
|  | * 1. Software Requirements |  |
|  | * 1. Software Description |  |
|  | * 1. Testing of Products |  |
|  | **LITERATURE SURVEY** |  |
|  | **CONCLUSION** |  |
| **7.** | **FUTURE** **ENHANCEMENT** |  |
| **8.** | **SAMPLE** **SCREENSHOT** |  |
| **9.** | **SAMPLE** **CODING** |  |
| **10.** | **REFERENCES** |  |

**ABSTRACT**

The scope of this work extends to system components (for example service providers, networks, servers, hosts, applications, processes and personnel) which are used to exchange PIN-related data. The PIN Guidelines in this document encompass PIN security within any one system or sub-system and between systems. This process designs 10 digit keypad with random RGB color SCHEME using a Fast Finite-State Algorithm for Generating RGB Palettes of Color. In this work, we propose a color finite-state LBG (CFSLBG) algorithm that reduces the computation time by exploiting the correlations of palette entries between the current and previous iterations

**CHAPTER 1**

**1. INTRODUCTION**

**1.1 General Introduction**

This work is designed to provide PIN security guidelines for all payment accounts that use a PIN, including those associated with magnetic stripe cards, chip cards, ‘hybrid’ cards that incorporate both a magnetic stripe and a chip or any other cardholder payment device form factor The scope of this work extends to system components (for example service providers, networks, servers, hosts, applications, processes and personnel) which are used to exchange PIN-related data. The PIN Guidelines in this document encompass PIN security within any one system or sub-system and between systems. These guidelines are targeted at PIN protection during PIN processing in the issuer security domain. PIN processing of interchange transactions is covered by the PCI PIN Security Requirements. A PIN (Personal Identification Number) is a four to twelve digit number known by a cardholder and used to authenticate the cardholder to the card-issuing bank (or to the cardholder’s ICC). The transaction PIN is the PIN entered by the cardholder during a payment transaction. The online PIN is the transaction PIN used to verify the cardholder online. The offline PIN is the transaction PIN used with an ICC to verify the cardholder offline. The reference PIN is a stored or derived PIN value used by the issuer to verify the transaction PIN. If stored in an ICC it may or may not be equal to the online PIN. The PIN management guidelines in this document refer to the following processes. The primary objective of this research paper is to provide a complete knowledge of ATM and a solution to its pin entry process. To access the funds which are kept in the bank at any time is not an easy task but today it is not at all difficult. A person just has to tell his bank that he wants an ATM card. The bank issues him an ATM card which is pass coded and could be used by him alone. ATM card is called by different names like bank card, MAC(Money access card), client card, key card or cash card, etc. In most cases even debit and credit card could be used. The ATM card helps the customer to be identified by a plastic ATM card with a magnetic stripe or a plastic smartcard with a chip. The security is provided by the customer entering a personal identification number (PIN). Now the person who needs funds when he is taking his family out of shopping or when he requires funds for an outing it is easily done. If someone falls very seriously ill the person can remove the funds in the middle of the night too. The ATM machine helps the customer to make his / her life much easier. It does most The ATM card is slowly replacing cheque, the personal attendance of various customers, has increased banking hours and reduced the holidays. They do not require any paper based verification. Due to heavier computing demands and the falling price of personal computer–like architectures, ATMs have moved away from custom hardware architectures using microcontrollers or application-specific integrated circuits and have adopted the hardware architecture of a personal computer, such as USB connections for peripherals, Ethernet and IP communications, and use personal computer operating systems

**1.2 OBJECTIVE**:

The main objective is,

* To perform the, Encryption and Decryption with less data loss.
* To implement the ATM learning algorithm.
* To enhance the performance analysis.

**1.3 PROBLEM STATEMENT**

* To perform the, Encryption and Decryption with less data loss.
* Although this method is straightforward and user, it has some severe limitations.
* Time taken to done the Encryption and decryption is very low, when compared with the other techniques

**CHAPTER 2**

**2. SYSTEM PROPOSAL**

**2.1 EXISTING SYSTEM**

* The use of a proxy server in a **Cloud Service Provider (CSP)** reduces search time and increases search efficiency by utilizing a **Boolean search** in the proxy server.
* Main server supports multiple users at a time with the help of Deep learning based Neural Network, which provides an accurate result.
* Trusted Authority is employed to provide secure document retrieval for authorized user. TA manages dual security processes as key management and Security Device Issuing.
* Secure top k ranking is achieved using Euclidean distance calculation and accuracy of document retrieval is improved.

**2.1.1 DISADVANTAGE**

* Encryption and decryption file on time high
* Along with that, data loss is more when compared with the other conventional methods.

**2.2 PROPOSED SYSTEM**

* Cloud services have increased the number of data owners it has been store their encrypted data in the cloud, while an equal or greater number of data users based in data retrieval.
* **ATM Algorithm** using the Encrypted and Decrypted the dataset
* Encrypted File will be Stored in Cloud Server and User based on Keyword Searching for Algorithm.
* User based Enter the keyword that also Encrypted Query After that Searching Encrypted Cloud Server
* Finally, deletion process is done to fetch the encrypted file, which is Related to the Query data.
* User based enter the Particular key user decrypts File the better performance better performance in terms of recall, ranking privacy, precision, searching time

**2.2.1 ADVANTAGE:**

* Time taken to done the Encryption and decryption is very low, when compared with the other techniques.
* Easy to retrieve the data from the cloud.
* Data loss is low, in the receiver side during the decryption process.

**CHAPTER 3**

**3. SYSTEM DIAGRAMS**

**3.1 ARCHITECTURE DIAGRAM**



**3.2 FLOW DIAGRAM:**

Input data

Data set (Cyber Security)

Handle missing value

BBC News

Dataset

Preprocessing

Label Encoding

**Dataset**

**ATM**

Cloud Storage

Encryption

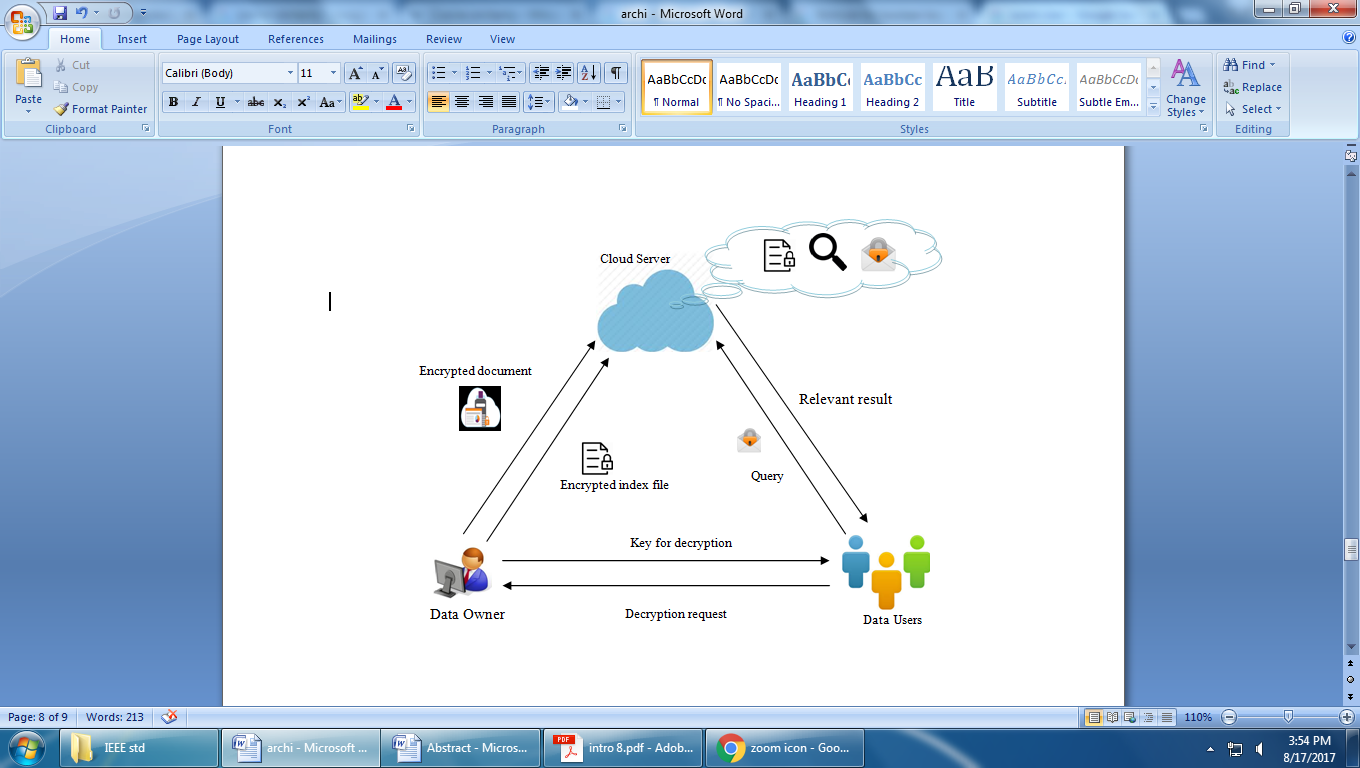
Decryption

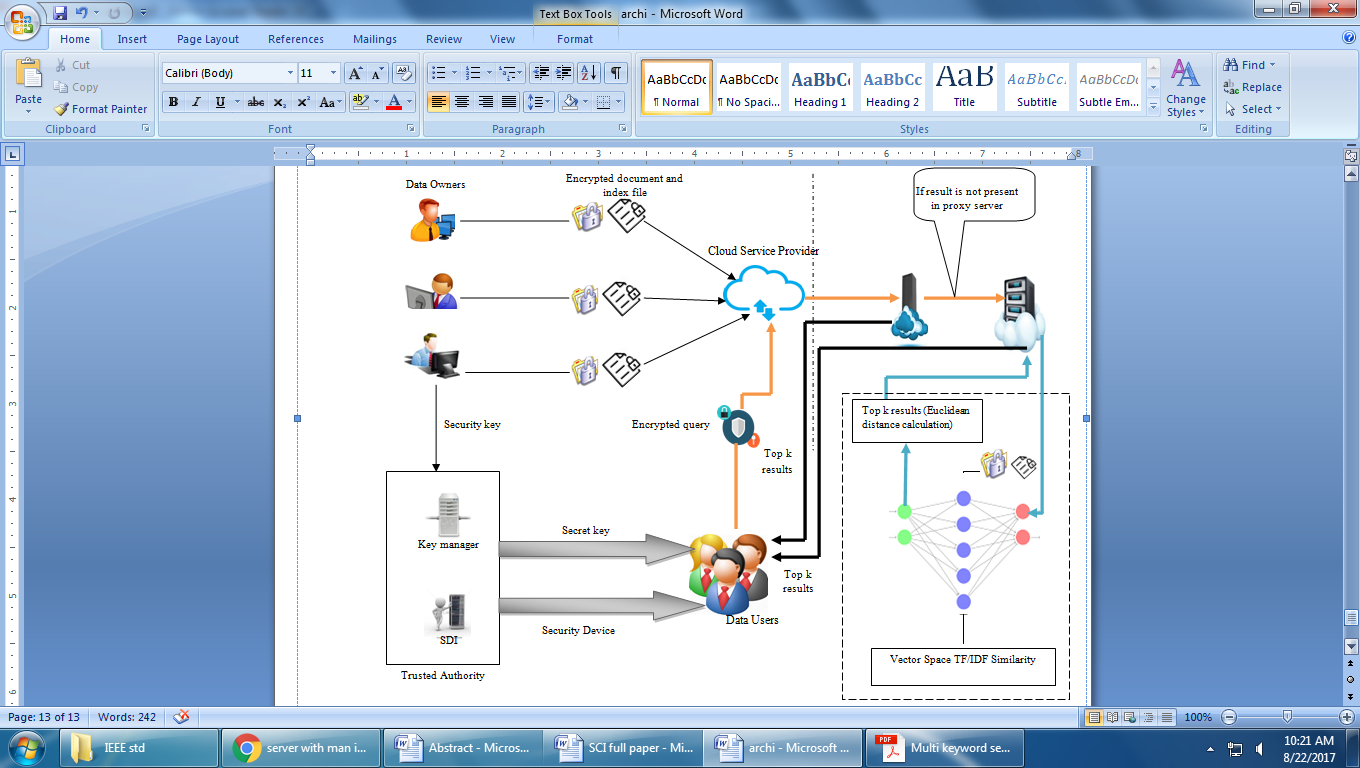
Query

Data deletion

Key

**3.3 BLOCK DIAGRAM**



****

**3.4 USE CASE DIAGRAM**

User

**CHAPTER 4**

**4. IMPLEMENTATION**

**4.1 MODULES**

* BBC News Dataset
* ATM Algorithm
* Query search
* Key Generation
* Cloud Storage AWS
* Performance Metrics

**4.1.1 DATASET**

* BBC NEWS is the use of technologies, processes, and controls to defend against cyber-attacks on systems, networks, programs, devices, and data.
* Its goal is to reduce the risk of BBC NEWS and to protect against unauthorized use of systems, networks, and technologies.
* BBC NEWS security Protocols Reference and Keywords
* In this step, we have to load the data with the help of panda’s packages.

**4.1.2 ATM Algorithm**

* ECC and AES based on Encrypted data
* The quick explanation is that keys using Elliptic Curve Cryptography (ECC) are asymmetric (public and private), whereas AES-256 uses a symmetric cypher (key)
* ECC and AES based on it Public and Private key

**4.1.3 Query Search**

* Hybrid AES and ECC based on **128 bit key** Generated For Encrypted data Wise

**4.1.4 Key Generation**

* ATM based on **128 bit key** Generated For Encrypted data Wise
* Which Can solve Security Problem itself
* Efficiently realize the information, data encryption, signature, and identity verification.

**4.1.5 Cloud Storage**

* Encrypted File Will be Stored in data for Security purpose
* A cloud computing model in which data is stored on the Internet via a cloud computing provider who manages and operates data storage as a service
* Cloud Server will be used on AWS Software

**CHAPTER 5**

**5. SYSTEM REQUIREMENTS**

The system requirement is the first step in the requirements analysis process. It lists the requirements of a particular software system including functional, performance and security requirements. The requirements also provide usage scenarios from a user, an operational and an administrative perspective. The purpose of software requirements specification is to provide a detailed overview of the software project, its parameters and goals. This describes the project target audience and its user interface, hardware and software requirements. It defines how the client, team and audience see the project and its functionality.

**HARDWARE AND SOFTWARE SPECIFICATION**

**5.1 HARDWARE SPECIFICATION:**

* System : Pentium IV 2.4 GHz
* Hard Disk : 200 GB
* Mouse : Logitech.
* Keyboard : 110 keys enhanced
* Ram : 4GB

**5.2 SOFTWARE SPECIFICATION:**

* O/S : Windows 7.
* Language : Python
* Front End : Anaconda Navigator - Spyder

**5.3 SOFTWARE DESCRIPTION**

**Python**

Python is one of those rare languages which can claim to be both *simple* and powerful. You will find yourself pleasantly surprised to see how easy it is to concentrate on the solution to the problem rather than the syntax and structure of the language you are programming in. The official introduction to Python is Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. I will discuss most of these features in more detail in the next section.

Features of Python

**Simple**

Python is a simple and minimalistic language. Reading a good Python program feels almost like reading English, although very strict English! This pseudo-code nature of Python is one of its greatest strengths. It allows you to concentrate on the solution to the problem rather than the language itself.

**Easy to Learn**

As you will see, Python is extremely easy to get started with. Python has an extraordinarily simple syntax, as already mentioned.

**Free and Open Source**

Python is an example of a FLOSS (Free/Libré and Open Source Software). In simple terms, you can freely distribute copies of this software, read its source code, make changes to it, and use pieces of it in new free programs. FLOSS is based on the concept of a community which shares knowledge. This is one of the reasons why Python is so good - it has been created and is constantly improved by a community who just want to see a better Python.

**High-level Language**

When you write programs in Python, you never need to bother about the low-level details such as managing the memory used by your program, etc.

**Portable**

Due to its open-source nature, Python has been ported to (i.e. changed to make it work on) many platforms. All your Python programs can work on any of these platforms without requiring any changes at all if you are careful enough to avoid any system-dependent features.

You can use Python on GNU/Linux, Windows, FreeBSD, Macintosh, Solaris, OS/2, Amiga, AROS, AS/400, BeOS, OS/390, z/OS, Palm OS, QNX, VMS, Psion, Acorn RISC OS, VxWorks, PlayStation, Sharp Zaurus, Windows CE and PocketPC!

You can even use a platform like Kivy to create games for your computer and for iPhone, iPad, and Android.

**Interpreted**

A program written in a compiled language like C or C++ is converted from the source language i.e. C or C++ into a language that is spoken by your computer (binary code i.e. 0s and 1s) using a compiler with various flags and options. When you run the program, the linker/loader software copies the program from hard disk to memory and starts running it.

Python, on the other hand, does not need compilation to binary. You just run the program directly from the source code. Internally, Python converts the source code into an intermediate form called bytecodes and then translates this into the native language of your computer and then runs it. All this, actually, makes using Python much easier since you don't have to worry about compiling the program, making sure that the proper libraries are linked and loaded, etc. This also makes your Python programs much more portable, since you can just copy your Python program onto another computer and it just works!

**Object Oriented**

Python supports procedure-oriented programming as well as object-oriented programming. In procedure-oriented languages, the program is built around procedures or functions which are nothing but reusable pieces of programs. In object-oriented languages, the program is built around objects which combine data and functionality. Python has a very powerful but simplistic way of doing OOP, especially when compared to big languages like C++ or Java.

**Extensible**

If you need a critical piece of code to run very fast or want to have some piece of algorithm not to be open, you can code that part of your program in C or C++ and then use it from your Python program.

**Embeddable**

You can embed Python within your C/C++ programs to give scripting capabilities for your program's users.

**Extensive Libraries**

The Python Standard Library is huge indeed. It can help you do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, FTP, email, XML, XML-RPC, HTML, WAV files, cryptography, GUI (graphical user interfaces), and other system-dependent stuff. Remember, all this is always available wherever Python is installed. This is called the Batteries Included philosophy of Python.

Besides the standard library, there are various other high-quality libraries which you can find at the Python Package Index.

**5.4 SYSTEM TESTING**

System testing is the stage of implementation, which aimed at ensuring that system works accurately and efficiently before the live operation commence. Testing is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an error. A successful test is one that answers a yet undiscovered error.

Testing is vital to the success of the system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved. . A series of tests are performed before the system is ready for the user acceptance testing. Any engineered product can be tested in one of the following ways. Knowing the specified function that a product has been designed to from, test can be conducted to demonstrate each function is fully operational. Knowing the internal working of a product, tests can be conducted to ensure that “al gears mesh”, that is the internal operation of the product performs according to the specification and all internal components have been adequately exercised.

**UNIT TESTING:**

Unit testing is the testing of each module and the integration of the overall system is done. Unit testing becomes verification efforts on the smallest unit of software design in the module. This is also known as ‘module testing’.

The modules of the system are tested separately. This testing is carried out during the programming itself. In this testing step, each model is found to be working satisfactorily as regard to the expected output from the module. There are some validation checks for the fields. For example, the validation check is done for verifying the data given by the user where both format and validity of the data entered is included. It is very easy to find error and debug the system.

**INTEGRATION TESTING:**

Data can be lost across an interface, one module can have an adverse effect on the other sub function, when combined, may not produce the desired major function. Integrated testing is systematic testing that can be done with sample data. The need for the integrated test is to find the overall system performance. There are two types of integration testing. They are:

1. Top-down integration testing.
2. Bottom-up integration testing.

**TESTING TECHNIQUES/STRATEGIES:**

**WHITEBOX TESTING:**

White Box testing is a test case design method that uses the control structure of the procedural design to drive cases. Using the white box testing methods, we

Derived test cases that guarantee that all independent paths within a module have been exercised at least once.

**BLACK BOX TESTING:**

* Black box testing is done to find incorrect or missing function
* Interface error
* Errors in external database access
* Performance errors.
* Initialization and termination errors

In ‘functional testing’, is performed to validate an application conforms to its specifications of correctly performs all its required functions. So this testing is also called ‘black box testing’. It tests the external behaviour of the system. Here the engineered product can be tested knowing the specified function that a product has been designed to perform, tests can be conducted to demonstrate that each function is fully operational.

**SOFTWARE TESTING STRATEGIES**

**VALIDATION TESTING:**

After the culmination of black box testing, software is completed assembly as a package, interfacing errors have been uncovered and corrected and final series of software validation tests begin validation testing can be defined as many,

But a single definition is that validation succeeds when the software functions in a manner that can be reasonably expected by the customer

**USER ACCEPTANCE TESTING:**

User acceptance of the system is the key factor for the success of the system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system at the time of developing changes whenever required.

**OUTPUT TESTING**:

After performing the validation testing, the next step is output asking the user about the format required testing of the proposed system, since no system could be useful if it does not produce the required output in the specific format. The output displayed or generated by the system under consideration. Here the output format is considered in two ways. One is screen and the other is printed format. The output format on the screen is found to be correct as the format was designed in the system phase according to the user needs. For the hard copy also output comes out as the specified requirements by the user. Hence the output testing does not result in any connection in the system

**CHAPTER 6**

1. **LITERATURE SURVEY**

* **Title** : Proficient Dual Secure Multi Keyword Search by Top k- Ranking based on Synonym Index and DNN in Untrusted Cloud
* **Year** : 2018
* **Author** : Aditi Gudadhe, Akanksha Parbat, Bhavana Wankhede, Brinda Darjee, Dr. Leena H.Patil
* **Methodology:**
* A secure ranking based multi keyword search using semantic index is being developed. Initially, owner builds an index file by semantic representation of keywords using Term Frequency/Inverse Document Frequency (TF/IDF). Security key is provided by Trusted Authority (TA) for decrypting the obtained results at the user side. TA manages dual security processes such as managing secret key and issuing security device to the data users. User query reaches proxy server, and it checks whether any frequent keyword matches with given query by Boolean Search. If not, query enters into the main server who stores all document and index files to obtain relevant result using Deep Learning Neural Network. In deep learning neural network, the query is processed with vector space model in order to retrieve the relevant documents. Finally, user decrypts the relevant results obtained from deep neural network. The experimental result shows that our proposed model provides better performance in terms of recall, ranking privacy, precision, searching time.

**Advantage:**

* encryption time and accuracy

**Disadvantage:**

* It requires basic information about keyboard shortcuts used or where the keys are located.

**2. LITERATURE SURVEY**

**Title:** Secure Ranked Multi-Keyword Search Based on Modified Blowfish algorithm and AVL Tree in Untrusted Cloud Environment

* **Year** : 2014
* **Author** : G.Shoba, G.Anusha, V.Jeevitha, R.Shanmathi
* **Methodology:**
* This MB algorithm provides robustness against any intruding whereas the conventional blowfish algorithm insecure for many applications. To achieve a proficient search, every data owner’s index based on AVL tree is encrypted by way of additive order and the privacy-preserving family is formed. The cloud server is then permitted to combine these indexes effectually without knowing the index content. An Iterative Deepening Depth First Search (IDDFS) procedure is used to discover the matching file for the data user request

**Advantage:**

* User query reaches proxy server, and it checks whether any frequent keyword matches with given query by Boolean Search.

**Disadvantage:**

* User query reaches proxy server, and it checks whether any frequent keyword matches with given query by Boolean Search is not matching on retrieval file.

**3. LITERATURE SURVEY**

# Title: Cyber Security Threats Detection in Internet of Things Using Deep Learning Approach

* **Year** : 2014
* **Author** : [Farhan Ullah](https://ieeexplore.ieee.org/author/37086246103)
* **Methodology:**
* This MB algorithm provides robustness against any intruding whereas the conventional blowfish algorithm insecure for many applications. To achieve a proficient search, every data owner’s index based on AVL tree is encrypted by way of additive order and the privacy-preserving family is formed. The cloud server is then permitted to combine these indexes effectually without knowing the index content. An Iterative Deepening Depth First Search (IDDFS) procedure is used to discover the matching file for the data user request

**Advantage:**

* More taken time for Encryption

**Disadvantage:**

* User Not matched Retrieval

**4. LITERATURE SURVEY**

# Title: Encrypted multi-keyword ranked search supporting gram based search technique

* **Year** : 2016
* **Author** : Suresh M
* **Methodology:**

Scheme that not only enable document keyword search but also supports linear, gram based and semantic searches. We construct a special tree-based index structure and propose a fuzzy Search Server that creates wild card based fuzzy keyword Set which overcome KGA (Keyword Guessing Attack) and provide efficient multi-keyword ranked search. KNN algorithm is used to encrypt the index and query. It also uses the relevance score calculation for ranking the documents.

**Advantage:**

* More taken time for Encryption

**Disadvantage:**

* It requires basic information about keyboard shortcuts used or where the keys are located.
* Relevance score calculation for ranking the documents Ranking Very low

**5. LITERATURE SURVEY**

# Title: Smart cloud search services: verifiable keyword-based semantic search over encrypted cloud data

* **Year** : 2016
* **Author** : Linga
* **Methodology:**

For protecting data privacy, sensitive data are always encrypted before being outsourced. Although the existing searchable encryption schemes enable users to search over encrypted data, these schemes support only exact keyword search, which greatly affects data usability. Moreover, these schemes do not support verifiability of search result. In order to save computation cost or download bandwidth, cloud server only conducts a fraction of search operation or return a part of result, which is viewed as selfish and semi-honest-but-curious. So, how to enhance flexibility of encrypted cloud data while supporting verifiability of search result is a big challenge. To tackle the challenge, a smart semantic search scheme is proposed in this paper, which returns not only the result of keyword-based exact match, but also the result of keyword-based semantic match. At the same time, the proposed scheme supports the verifiability of search result..

**Advantage:**

* Less taken time for Encryption

**Disadvantage:**

* It requires basic information about keyword-based semantic match not perfect matching.
* Relevance score calculation for ranking the documents Ranking Very low
* More taken time for Semantic Searching

**6. LITERATURE SURVEY**

# Title: Privacy preserving synonym based fuzzy multi-keyword ranked search over encrypted cloud data

* **Year** : 2014
* **Author** : Linga
* **Methodology:**

Privacy Preserving Synonym Based Fuzzy Multi-Keyword Ranked Search over Encrypted Cloud Data, a scheme which enhances user search experience to a paramount by providing both fuzzy and synonym based multi-keyword ranked search, thereby taking encrypted search experience closer to free text search engines. The scheme additionally improves upon index generation time and search time in comparison to existing schemes by utilizing a binary tree based dynamic index. Experimental results portray the effectiveness of this proposed scheme as it reduces the search time

**Advantage:**

* As it reduces the search time
* **Disadvantage:**
* It is Convert File the data huge time taken For binary tree based dynamic index
* More taken time for Semantic Searching

**CHAPTER 7**

**CONCLUSION**

* **Hybrid ECC and AES Algorithm** using the Encrypted and Decrypted the dataset
* Encrypted File will be Stored in Cloud Server and User based on Semantic Searching method Algorithm.
* User based keyword Entering is done to retrieve the corresponding data file from the cloud storage.
* Finally Retrieve the Related File based on Query.
* This will easily Find out **Cyber security Problem** like, the fault file will be detected

**CHAPTER 8**

**FUTURE ENHANCEMENT**

* It is based on Future Hybrid algorithm for ECC and Diffieleman Algorithm
* You may increase the Efficiency of the Algorithm.
* we Can future implementation on testing Software for Cyber Security

**CHAPTER 9**

**CHAPTER 11**

**REFERENCE PAPER**

1. L. Zhang, Y. Zhang and H. Ma, “Privacy-Preserving and Dynamic Multi-Attribute Conjunctive Keyword Search Over Encrypted Cloud Data”, IEEE Access, vol. 6, pp. 34214-34225, 2018.
2. Z. Xiangyang, D. Hua, Y. Xun, Y. Geng, and L. Xiao, “MUSE: An Efficient and Accurate Verifiable Privacy-Preserving Multi-keyword Text Search over Encrypted Cloud Data”, Security and Communication Networks, vol. 2017, pp. 1-17, 2017.
3. L. Chen, L. Qiu, K-C. Li, W. Shi, and N. Zhang, "DMRS: an efficient dynamic multi-keyword ranked search over encrypted cloud data", [Soft Computing](https://link.springer.com/journal/500), vol. 21([16](https://link.springer.com/journal/500/21/16/page/1)), pp. 4829–4841, 2017.
4. R. Zhang, R. Xue, L. Liu, and L. Zheng, “Oblivious Multi-Keyword Search for Secure Cloud Storage Service”,2017 IEEE 24th International Conference on Web Services, pp. 269-276, 2017.
5. P. K. Samantaray, N. K. Randhawa, and S. L. Pati, “An Efficient Multi-keyword Text Search Over Outsourced Encrypted Cloud Data with Ranked Results”, [Computational Intelligence in Data Mining](https://link.springer.com/book/10.1007/978-981-10-8055-5), pp. 31-40, 2018.
6. Z. Fu, F. Huang, K. Ren, J. Weng, and C. Wang, “Privacy-Preserving Smart Semantic Search Based on Conceptual Graphs Over Encrypted Outsourced Data”,IEEE Transactions On Information Forensics And Security, vol. 12(8), pp. 1874-1884, 2017.
7. B. Lang, J. Wang, M. Li, and Y. Liu, “Semantic-based Compound Keyword Search over Encrypted Cloud Data”, IEEE Transactions On Services Computing, 2018.
8. Z. Fu, L. Xia, X. Sun, A. X. Liu, and G. Xie, “Semantic-aware Searching over Encrypted Data for Cloud Computing”, I[EEE Transactions on Information Forensics and Security](https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=10206), vol. 13(9), pp. 2359-2371, 2018.
9. Z. Wu, and K. Li, “VBTree: forward secure conjunctive queries over encrypted data for cloud computing”, [The VLDB Journal](https://link.springer.com/journal/778), pp. 1–22, 2018.
10. Y. Yang, X. Liu, and R. H. Deng, “Multi-user Multi-Keyword Rank Search over Encrypted Data in Arbitrary Language”, IEEE Transactions on Dependable and Secure Computing, 2018.
11. X. Ding, P. Liu, and H. Jin, “Privacy-Preserving Multi-keyword Top-*k* Similarity Search Over Encrypted Data”,IEEE Transactions on Dependable and Secure Computing, 2018.
12. Y. Li, F. Zhou, Y. Qin, M. Lin, and Z. Xu, “Integrity-verifiable conjunctive keyword searchable encryption in cloud storage”, [International Journal of Information Security](https://link.springer.com/journal/10207), vol. 17[(5](https://link.springer.com/journal/10207/17/5/page/1)), pp. 549–568, 2018.
13. C. Guo, X. Chen, Y. Jie, Z. Fu, M. Li, and B. Feng, “Dynamic Multi-phrase Ranked Search over Encrypted Data with Symmetric Searchable Encryption”, IEEE Transactions On Services Computing, 2018.
14. Raghavendra S, Girish S, Geeta C. M., R. Buyya, Venugopal K. R., S. S. Iyengar, and L. M. Patnaik, “Split keyword fuzzy and synonym search over encrypted cloud data”, [Multimedia Tools and Applications](https://link.springer.com/journal/11042), vol. 77([8](https://link.springer.com/journal/11042/77/8/page/1)), pp. 10135–10156, 2018.
15. A. V. Vora, and S. Hegde, “Keyword-based private searching on cloud data along with keyword association and dissociation using cuckoo filter”, [International Journal of Information Security](https://link.springer.com/journal/10207), pp. 1–15, 2018.
16. H. Wang, X. Dong, and Z. Cao, “Secure and efficient encrypted keyword search for multi-user setting in cloud computing”, [Peer-to-Peer Networking and Applications](https://link.springer.com/journal/12083), pp. 1–11, 2018.
17. Z. Shen, J. Shu, and W. Xue, “Keyword Search with Access Control over Encrypted Cloud Data”, IEEE Sensors Journal, vol. 17(3), pp. 858-868, 2017.
18. N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, “Privacy-Preserving Multi-Keyword Ranked Search overEncrypted Cloud Data,” Proc. IEEE INFOCOM, pp. 829-837,   
    Apr,2014.
19. Jiadi Yu, Peng Lu, Yanmin Zhu, Guangtao Xue, Member, IEEE Computer Society, and Minglu Li, ”Toward Secure Multikeyword Top k Retrieval over Encrypted Cloud Data”, IEEE Transactions, July 2013.
20. Ming Li et al.,” Authorized Private Keyword Search over Encrypted Data in Cloud Computing, IEEE proc. International conference on distributed computing systems, June 2011, pages 383-392.
21. J. Li et al., “Fuzzy Keyword Search Over Encrypted Data in Cloud Computing,” Proc. IEEE INFOCOM, 2010, pp. 441–45.
22. Ming Li et al., ”Toward Privacy-Assured and Searchable Cloud Data Storage Services”, IEEE Transactions on Network, volume 27, Issue 4, July/August2013.
23. D. Boneh, G.D. Crescenzo, R. Ostrovsky, and G. Persiano, “Public Key Encryption with Keyword Search,” Proc. Int’l Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT), 2004.
24. Zhou et al., “K-Gram Based Fuzzy Keyword Search over Encrypted Cloud Computing “Journal of Software Engineering and Applications, Scientific Research , Issue 6, Volume 29-32,January2013.
25. E.-J. Goh, “Secure Indexes,” Cryptology ePrint Archive, http://eprint.iacr.org/2003/216.2003.
26. M. Li, S. Yu, N. Cao, and W. Lou, “Authorized Private Keyword Search over Encrypted Data in Cloud Computing,” Proc. 31st Int’l Conf. Distributed Computing Systems (ICDCS ’10), pp. 383-392, June 2011.
27. C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, “Secure Ranked Keyword Search over Encrypted Cloud Data,” Proc. IEEE 30th Int’l Conf. Distributed Computing Systems (ICDCS’10), 2010.