International Journal of Advance Research in Science and Engineering
Vol. No.5, Issue No. 12, December 2016
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

FULLY HOMOMORPHIC ENCRYPTION WITH RELATIVELY SMALL KEY AND CIPHER TEXT SIZES

Mr. S. Jagadeesan

Research Scholar, SRM University, Chennai (India)

ABSTRACT

To transmit private searching on streaming data is a process to a public server a program which searches streaming sources of data without helpful searching criteria and sends back a buffer containing the findings. The searching criteria can be constructed by only simple combination of keywords, for example disjunction of keywords. The breakthrough is recent in comparison based computing has allowed to construct random searching criteria. A new private query searches for documents from data streaming on the basis of keyword frequency. The keyword frequency is required to be higher or lower than a given threshold. This query can help us finding more relevant documents based on the state of the art fully homomorphic encryption techniques, disjunctive, conjunctive and complement constructions for private threshold queries based on query searching. The proposed solution is to search for documents containing more than t keywords out of n and threshold searching without increasing the size of the dictionary. The fully comparison based computing technique and the buffer keeps at most k matching documents without collisions. The documents searching containing one or more classified keywords threshold searching. Now consider a new private query which searches for documents from streaming data based query searching such that a number of times that a keyword appears in a matching document are required to be higher or lower than a given threshold.

Keywords : FF (file format), streaming, Threshold, Filter generation

I. INTRODUCTION

The document search containing more keywords (k) out of n keywords is called (k; n) threshold searching. The Comparison Based Computations technique and buffer keeps at most m matching documents without crash finding for documents having one or more classified keywords like can be achieved by (1; n) threshold probing. New query which finds for documents from flowing data based on keyword on availability. The most of the times a keyword appears in a matching document is not required to be higher or lower than a given value of the original value. Another method is proposed for taking matching documents from the buffer. The streaming data was motivated by one of the tasks of the intelligence community using private searching and how to collect useful information potentially from huge volumes of streaming data flowing through a public server. A red flag is often classified and

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

satisfies secret search. Our aim to face up how to keep the search top secret even if program reside in the public server falls into adversary's hands. There are many applications problem purpose of intelligence gathering.

The private stream searching allows a client to retrieve the identical documents using search criteria from the remote server while the server evaluating the request to the search criteria. The abstract extensive to give a high altitude of a new scheme for this problem and analysis of its scalability. The innovative plan is highly to demonstrate the practical applicability of the scheme considering its routine in the exact state of providing a privacy preserve version of the Google news ready to act service. The idea of an encrypted dictionary is used and no matching documents have no effect on the contents of the large buffer and attempt to avoid crash. The time execution of the evaluation function with respect to the value of integer bit length is illustrate the time value of even reason are normalize. The new curve is very close to the theoretical given as exciting example; this is to gather mostly useful information from enormous streaming sources of data. The huge information is available and used to store all the standard informations are not useful. In online method data is normally sending from multiple data streams and only single information packet at a time. The huge amount of the data is immediately dismiss and drop at the same time a little amount of likely helpful information is taken. The flowing data sources are given by a quantity of network routers for packet entering. The red flags increase to collect a small subset of data for analysis in a secure environment. A limited number of operations are performed for Partial homomorphic scheme that is either addition or multiplication on encrypted data due to a lack of ability. A few partial homomorphic cryptosystems are Unpadded RSA, Goldwasser-Micali, ElGamal, Benaloh, Paillier, Okamato-Uchiyama, Naccache-Stern, Damgard-Jurik, and Boneh-Goh-Nissim.

Fully homomorphic encryption allows a worker to receive encrypted data and perform complex operation in spite of not having the secret decryption key. FHE performs both addition and subtraction operations on encrypted data without affecting the structure of the plaintexts. A homomorphic encryption scheme allows any to publicly transform a collection of cipher texts for some plaintext. A user can store encrypted data on a server and allow the server to process the encrypted data without revealing the data to the server. These schemes supported only a limited set of functions which restricted their applicability. The different applications of FHE are related to research works are

Damgard et al: based on homomorphic encryption and does not use random oracles

Brenner et al: solving problems of encrypted storage access with encrypted addresses and encrypted branching and comprise of the runtime environment for an encrypted program and an assembler to generate the encrypted machine code.

Gahi et al: to measure the time a query on the database takes to execute

Wei and Reiter: a client to evaluate an encrypted file stored at a server using deterministic finite automation

GHS: implemented leveled homomorphic encryption for evaluation of AES-128 circuit

Bonneh et al: to control the number of homomorphic computations one can perform on encrypted data.

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

Gahi et al: allowing users to benefit from location based services not compromising the data confidentiality and integrity.

Gennaro and Wilch: a symmetric key variation of fully homomorphic signatures

Wang et al: introduced an array of algorithmic optimizations

Yukun et al: message digest and digital signature are also provided.

Gauraha et al: fulfills I/O confidentiality, duplicitous pliability and competence.

Mani et al: query processing on an encrypted database.

Gupta et al: symmetric keys based on matrix operations

Joo and Yun: Secure against ciphertexts attacks both for privacy and authenticity and is based on the error-free approximate GCD assumption.

Yuan and Yu: each party to encrypt his/her private data locally followed by uploading the ciphertexts into the cloud.

Vijay and Sharma: The ciphertexts accumulate noise after a certain number of computations,

II RELATED WORK

The several researches are being carried out to enhance the private query and also satisfy the secret search. The semantic security using the lesser keywords for data searching and some research works are motivated us to develop our future system. The several problems related to private searching include the data is encrypted and the query is unencrypted [3, 4], the single database retrieve private information retrieval (PIR) [5, 6], and the problem is closely relative. The private searching problem as defined above and only require the number of the same documents shows the communication dependent. The drawback is that steep resource requirements that limit its practical application for many of the scenarios described above. Keywords of each query to be selected from a public, unencrypted dictionary. In many applications, including a user's search keywords in the public dictionary will already reveal too much information about the client's interests Such schemes are well known to be useful for constructing privacy-preserving protocols, A client can enlarge the host of encrypted data, and permits the host to progression the encrypted data without revealing the data to the host. Encryption techniques supported only a limited set of functions *f*, which limited their availability.

In private searching a client will create an encrypted query for the set of keywords. The user will provide this encrypted request for information to the host. The host will then run a look for algorithm on a stream of files while keeping an encrypted buffer storing information about files for which there is a keyword contest. The encrypted register will then be return to the client to enable the client to rebuild the files that enquiry keywords that are equal. If file is related file it matches at least one keyword in the set of keywords that the client is involved in. a confidential searching method is able to conducting the search even though it does not know which set of keywords the client is interested in.

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

Fully homomorphic encryption (FHE) allows a worker to take encrypted data and perform arbitrarily-complex dynamically-chosen computations on that data while it leftovers encrypted. All FHE methods followed the unique design technique, one method in Gentry's unique construction [8, 7]. The first step in Gentry's design is to construct a somewhat homomorphic encryption (SWHE) scheme, and it was called namely as encryption scheme capable for evaluating low degree polynomials initial with based on similar lattice [8]. The lattices are based on either directly or absolutely. The methods of cipher texts are really noisy, for using these methods with some noises that grows a little for during homomorphic addition. The middle of during homomorphic multiplication, and the restriction of low-degree polynomial times. The main aim is to obtain encryption technique; Gentry provided a extraordinary bootstrapping method which states that given a SWHE scheme that can evaluate its own decryption function, The level 1 of the FHE method is called by the bootstrapping. By adapting the Gentry Bootstrapping that refreshes the cipher text by homomorphically running the decryption function, The encrypted secret key is used in public key of the user and then resulting in a reduced noise in the data of they given. And it is used by a strange rule of life and it so, to obtain the Homomorphic Techniques are likely to be not capable of solving their own decryption circuits without significant changes. We have seen the final step is to crush the decryption circuit of the SWHE scheme, namely transform the scheme into one with the same homomorphic capacity but a decryption circuit that is simple enough bootstrapping is allowed by the method of the circuit. Those were really under this adding a hint showed by Gentry method, for a large set with a preceding the secret sparse subset and that sums to the original secret key to the public key and relying on a "sparse subset sum" assumption.

The efficiency of fully homomorphic encryption is used by the different scenarios, in fact the question following its finding. Here it is considered with the per-gate computation overhead of the FHE scheme, defined as the ratio between the time it takes to compute a circuit homomorphically to the time it takes to compute it in the very close up manner. Unluckily, The FHE methods that follow Gentry's blueprint some of which have actually been implemented have fairly poor performance of their per-gate computation a large polynomial in the security limitation. In this they are going to like to argue that it is penalty in performance is somewhat inherent for schemes that follow this blueprint. First, the complexity bootstrapping is inherently at least the complexity of decryption times the bit-length of the individual cipher texts that are used to encrypt the bits of the secret key. The main thing happen here is that bootstrapping involves an evaluating the decryption circuit homomorphically it is, closed in the decryption circuit, each secret-key bit is replaced by a cipher text that encrypts that bit.

Very informally, given a program f from a class C of programs, and a security parameter k, a public-key program obfuscator compiles f into (F;Dec), where F on any input computes an encryption of what f would compute on the same input for encryption. And then using the decryption algorithm Dec decrypts the output of F. That is, for any input x, Dec (F(x)) = f(x), and in really the given code for F it is impossible to distinguish for any polynomial time adversary which f from the class C was used to produce F. We stress that in our definition, the program encoding length jFj must polynomially depend only on jfj and k, and the output length of jF(x)j are must be polynomial

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

depend only on jf(x)j and k. It is easy to see that Single-Database Private Information Retrieval (including keyword search) can be viewed as a special case of public-key program obfuscation.

Here how to public-key program obfuscate keyword search algorithms on streaming data, where the size of the query (i.e. complied executable) must be independent of the size of stream (i.e., database), and that can be executed in an online environment, That is nothing but one document at a time. And this results also can be viewed as improvement and a speedup of the best previous results of single round PIR with keyword search according to the way of the introduction of the streaming model, And also it improves the preceding work on keyword PIR by allowing for the simultaneous return of multiple documents that match a set of keywords, There is a ability to more competently perform different types of queries beyond just searching for a single keyword. It is showed how to collect the disjunction of a set of keywords and several other functions.

Homomorphic encryption

A solution to this problem is homomorphic encryption, which permits computing on encrypted data. That is, the client can encrypt his data x and send the encryption Enc(x) to the server. The server can then take the cipher text Enc (x) and evaluate a function f on the underlying x obtaining the encrypted result Enc (f(x)).

III ARCHITECTURE

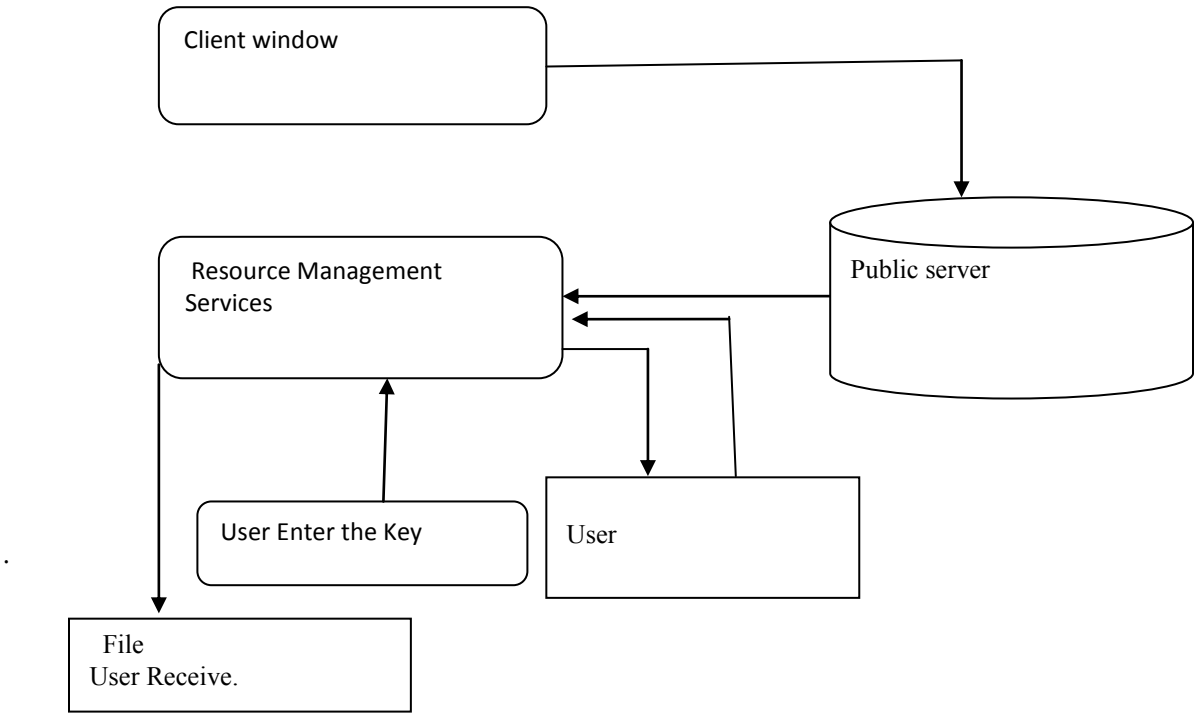
Straightforward implementation of many machine learning algorithms requires operations which are not necessarily represented by a low-degree polynomial such as comparison and division, making difficult to adapt certain algorithms to operate on encrypted data. For instance, a comparison $x > y$ is not polynomial time value, It seems unless the values of inputs are encrypted bit-wise and a deep circuit for comparison is implemented. Solving this problem may lead to practical implementations. The rest of this article is organized as follows: first, we present the main definitions and notations used overall here. And show some of the schemes developed after Gentry's recent break in, all its solutions are focused on the method that was really implemented in the form of a software library called lib. In the second part of the paper, we model the numbers comparison in terms of encryption, and test this approach using lib. Finally, Here discussing with a comparison-based computation method, And the maximum value from a vector.

3.1 Mixture File Generation

These windows are used to send a message from one peer to another. We use the Swing package available in Java to design the User Interface. Swing is a widget toolkit for Java. It is part of Sun Microsystems' Java Foundation Classes an API for providing a graphical user interface for Java programs. In this module mainly we are focusing the login design page with the Partial knowledge information Here always see the result of the mix of all these components in the one file Continuous aggregation queries over dynamic data are used for real time decision making and timely business intelligence. We consider queries where a client wants to be notified over distributed data crosses a specified file.

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

3.2 Threshold Quieries



.

Fig 3.1 General Architecture Diagram

The performance comparison of our threshold query protocols can be summarized in Complexity client and Complexity server, where enc and dec stand for encryption and decryption of bit, add and multi denote the homomorphism addition and multiplication of bits, and ADD represents the homomorphism. The performance of our generic construction depends on the performance of the underlying basic constructions addition of integers.

3.3 Semantic Security

Databases systems are central to most organizations' information systems strategies. In an organizational level, users can expect to have frequent contact with database systems. Therefore, skill in using such systems understanding their capability and limits, knowing how to access data directly or through technical specialists. Semantic security provides measures for preventing, detaining or minimizing effects of semantic attacks. Traditional approaches to information system security focused on protecting systems and the information stored, processed and distributed on them. The goal of this project is to develop techniques to detect inconsistencies or irregularities (Behaviour that breaches the rule, custom or morality) in online information.

IV CONCLUSION AND FUTURE ENHANCEMENTS

The innovative architecture that guarantees confidentiality of data stored in public databases .And the state of the

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

art Comparison-Based Computations techniques were based on this, we have presented constructions Performance of number of Comparisons for disjunctive, conjunctive, and complement required queries based on keyword frequency and then a construction for a generic threshold query based on keyword frequency. These methods are semantically secure as long as the underlying Comparison-Based Computations scheme is semantically secure. For this construction for disjunctive threshold query is able to search for documents. In future, an innovative architecture that guarantees confidentiality of data stored in public database. To improve the performance of construction can post process the cipher text of a bit in the final stage. Any search criteria can be constructed with fully homomorphic encryption scheme in private searching on streaming data and different queries will need different constructions.

REFERENCES

[1] C. Gentry, Computing Arbitrary Functions of Encrypted Data, Comm. ACM, vol. 53, no. 3, pp. 97-105, 2010.

[2] C. Gentry, Toward Basing Fully Homomorphic Encryption on Worst-Case Hardness, Proc. Advances in Cryptology (CRYPTO ’10), pp. 116-137, 2010.

[3] C. Gentry and S. Halevi, Implementing Gentry’s Fully-Homomorphic Encryption Scheme, Proc. 30th Ann. Int’l Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT ’11), pp. 129-148, 2011.

[4] D. Harris, D.M. Harris, and S.L. Harris, Digital Design and Computer Architecture. Morgan Kaufmann, 2007.

[5] D.J. Lilja and S.S. Sapatnekar, Designing Digital Computer Systems with Verilog. Cambridge Univ. Press, 2005.

[6] S. Ling and C.P. Xing, Coding Theory: A First Course. Cambridge Press, 2004.

[7] R. Ostrovsky and W. Skeith, Private Searching on Streaming Data, Proc. Advances in Cryptology (CRYPTO ’05), pp. 223-240,2005.

[8] R. Ostrovsky and W. Skeith, Private Searching on Streaming Data, J. Cryptology, vol. 20, no. 4, pp. 397-430, 2007.

[9]R. Ostrovsky and W. Skeith, Algebraic Lower Bounds for Computing on Encrypted Data, Proc. Electronic Colloquium on Computational Complexity (ECCC ’07), 2007.

[10] P. Paillier, Public Key Cryptosystems Based on Composite Degree Residue Classes, Proc. 17th Int’l Conf. Theory and Application of Cryptographic Techniques (EUROCRYPT ’99),pp. 223-238, 1999.

[11] B. Parhami, Computer Arithmetic: Algorithms and Hardware Designs, second ed. Oxford Univ. Press, 2010.

[12]N. Smart and F. Vercauteren, Fully Homomorphic Encryption with Relatively Small Key and Cipher text Sizes, Proc. 13th Int’l Conf. Practice and Theory in Public Key Cryptography (PKC ’10),pp. 420-443, 2010.

[13]D. Stehle and R. Steinfeld, Faster Fully Homomorphic Encryption, Proc. Advances in Cryptology (ASIACRYPT ’10), pp. 377-394,2010.

[14]J.F. Wakerley, Digital Design Principles and Practices, third ed. Prentice Hall, 2000.