International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Homomorphic Encryption Method for Bio Medical Systems

Dr. Gorti V N K V Subbarao[1], Dr. Gorti V Karthikeya[2], Dr. Gorti V Viswaq[3]

[1]Professor, CSE, Neil Goggty Institute of Technology
[2,3]PG Student, Medical, SGMC

Abstract:
Globally accessible databanks contain diagnostic results for patients with chronic conditions such as diabetes and hypertension. These resources provide valuable insights, including susceptibility to various diseases and familial relationships. However, the sensitive nature of this data necessitates privacy, as parties may prefer not to disclose their information to one another. Therefore, a protocol or technique is essential to ensure confidentiality. We are introducing an Enhanced Homomorphic Cryptosystem designed to fulfill industry standards. In this context, we will implement our scheme alongside other homomorphic encryption methods to evaluate the data in ciphertext form, thereby safeguarding privacy. This discussion will focus on the new protocol for healthcare systems utilizing Enhanced Homomorphic Cryptosystems, while also comparing it to existing schemes. Additionally, we will analyze the computational costs associated with several homomorphic encryption methods in relation to our new approach.

Keywords: Enhanced Homomorphic Cryptosystem (EHC), Electronic medical records (EMR), Mixed Multiplicative Homomorphism (MMH),Elgamal,Security,Power Consumption.

1. Introduction

Over the past decade, it has become a common practice for organizations to delegate their online business operations to web hosting service providers. Typically, third-party hosts manage both the databases and business logic of a company to reduce IT management time and costs. This trend has been further propelled by cloud computing services. Numerous cloud-based data centers now store vast amounts of data from various sources and facilitate data-centric computations. However, security remains a significant concern for such hosted data, particularly when it involves sensitive information. There is a risk that a data center could be subjected to attacks and breaches. Additionally, the possibility of insider threats exists. Changes in management, such as reorganizations or restructures, can heighten this risk due to increased exposure to multiple management personnel and the lack of established policies for handling critical information in these scenarios. While encrypting critical data can address some security issues associated with outsourced databases, it introduces the challenge of how data centers can perform computations on encrypted data. This challenge can be mitigated through the use of homomorphic encryption, which has been extensively researched over the past thirty years.

The increasing demand for digital data privacy, along with the use of communication networks and electronic devices, is paralleled by a rise in attacks on electronic goods and sensitive information. These attacks can involve manipulation, destruction, or theft of data. To securely store and access data, various methods exist to ensure privacy, including encryption and tamper-resistant hardware. The challenge intensifies when considering the ability to compute encrypted private data publicly without compromising its integrity or revealing it to unauthorized parties. The solution to this challenge lies in homomorphic encryption. Furthermore, in everyday communication and data storage, confidentiality is upheld through encryption. In modern work environments, it is essential to implement functionalities such as delegating computations and comparing data with untrusted nodes or organizations for further processing. The only viable approach is to provide data in an encrypted format for subsequent processing. In 1978, Rivest et al. proposed a solution to this issue through homomorphic encryption, which allows operations to be conducted on encrypted data while preserving confidentiality. Homomorphic encryption schemes enable specific operations to be executed on encrypted data as if they were performed on the original plaintext.

2. Homomorphic encryption scheme

Homomorphic encryption is a technique that enables complex computations to be executed on encrypted data. It is characterized as a unique encryption method that permits operations on encrypted data as if they were conducted on the original plaintext. This type of encryption can be categorized into additive, subtractive, multiplicative, mixed multiplicative, and mixed addition. In additive homomorphism, decrypting the sum of two ciphertexts yields the same result as adding the two plaintexts, expressed as $E(x+y) = E(x) + E(y)$. In subtractive homomorphism, decrypting the difference between two ciphertexts corresponds to subtracting the two plaintexts, represented as $E(x-y) = E(x) – E(y)$. In multiplicative homomorphism, decrypting the product of two ciphertexts is equivalent to multiplying the two plaintexts, denoted as $E(x*y) = E(x) * E(y)$. Mixed multiplicative homomorphism allows for the decryption of the product of one ciphertext and one plaintext to equal the encryption of their plaintext product, shown as $E(x*y) = E(x) * y$. While this property can be problematic as it may expose information by compromising the encryption, it is advantageous when one needs to verify the sum of a set of encrypted values without disclosing the values themselves. This principle is applied in voting protocols to confirm the tally of votes without revealing the individual ballots.

2.1.There are several efficient Homomorphic cryptosystems [21] :

The concept of encryption methods that allow complex computations on encrypted data was initially introduced by Rivest, Adleman, and Dertouzos in 1978 in a prescient paper titled 'On Data Banks and Privacy Homomorphisms.' In this work, they suggested the use of the exponentiation function and the RSA function as additive and multiplicative privacy homomorphisms, respectively. It is important to note that neither function alone offers chosen plaintext security. However, attempts to adapt the RSA function into a CPA-secure encryption scheme, through either the hardcore-bit construction or the RSA-OAEP construction, appear to compromise its homomorphic capabilities. While the RSA cryptosystem is homomorphic with respect to multiplication, it is evident that this property does not extend to addition, indicating that RSA is not algebraically homomorphic. Furthermore, the RSA cryptosystem does not achieve IND-CPA security unless messages are padded randomly. In contrast, the Benaloh cryptosystem facilitates the homomorphic addition and subtraction of ciphertexts.

The initial semantically secure homomorphic encryption scheme is based on the foundational work of Goldwasser and Micali, who established the first comprehensive security framework for encryption. The GM encryption scheme facilitates the addition of encrypted bits modulo 2, effectively implementing the exclusive-OR function. Other additively homomorphic encryption schemes that have been proven to be semantically secure include Benaloh, Naccache-Stern, Okamoto-Uchiyama, Paillier, and Damgard-Jurik. Following these developments, several encryption systems exhibiting either additive or multiplicative homomorphism emerged, including the ElGamal encryption scheme. This scheme, which is based on the exponentiation function, is characterized as multiplicatively homomorphic and CPA-secure. The ElGamal cryptosystem enables the homomorphic multiplication of two encrypted messages, as well as multiplication and exponentiation by known constants. A threshold construction of the ElGamal cryptosystem has been proposed by Cramer, Gennaro, and Schoenmakers, who also introduced a new variant of ElGamal that is additively homomorphic. The Paillier encryption scheme and its generalization by Damgard and Jurik, along with a variety of lattice-based encryption schemes originating from the work of Ajtai and Dwork, further contribute to this field. The Naccache-Stern cryptosystem allows for the homomorphic addition and subtraction of ciphertexts, in addition to multiplying a ciphertext by a constant. The Okamoto-Uchiyama cryptosystem supports the homomorphic addition and subtraction of two ciphertexts, as well as addition and multiplication by a known constant, and provides efficient re-randomization of ciphertexts. The Paillier cryptosystem enables homomorphic addition and subtraction of encrypted messages, along with the addition and subtraction of constants, and multiplication by a constant. The Damgard-Jurik cryptosystems inherit all the homomorphic properties of the Paillier cryptosystem.

3. Applications of Homomorphic Encryption Scheme

Protocols for securely comparing private data in Bioinformatics include threshold schemes such as secret sharing, mobile agent protection, multiparty computation, and multi-agent dynamic programming. These protocols ensure mobile agent security, anonymity, and privacy in applications like e-voting, electronic auctions, and lottery systems. Additionally, they incorporate zero-knowledge proofs, commitment schemes, mix-nets, and watermarking or fingerprinting protocols. Security solutions extend to wireless sensor networks and secure minimum optimal path finding in computer networks, as well as secure packet forwarding in mobile ad hoc networks. Other relevant concepts include threshold cryptography, oblivious transfer, and secure set intersection protocols for databases and cloud computing.

4. Motivation

Rivest, Adleman, and Dertouzos [RAD78] introduced the concept of privacy homomorphism. However, it wasn't until 2009 that Gentry [Gen09] developed the first fully homomorphic encryption based on ideal lattices, as previous methods lacked complete security. Following Gentry's breakthrough, there has been significant interest in fully homomorphic encryption (FHC). In FHC, an arbitrary ciphertext is generally represented as $C = qp + 2r + m$ [28]. A straightforward attack involves eliminating 'qp' from the ciphertext C by introducing a small noise value, which facilitates the recovery of the plaintext bit m. This attack's success is attributed to the secret key p being a large integer. While existing schemes typically support either homomorphic addition or multiplication of plaintexts, our scheme accommodates both operations, including mixed addition and multiplication. A review of the literature

reveals several efficient partially homomorphic cryptosystems and two fully homomorphic but less efficient ones. Although unintentionally homomorphic cryptosystems may be vulnerable to attacks, careful handling of homomorphism can enable secure computations. For example, one individual can add two encrypted numbers, and another can decrypt the result without either party discovering the individual values.

4.1.Need for improvement

The primary features of encryption primitives include the algorithm, performance, and security. Numerous encryption schemes documented in literature fulfill some or all of the specified criteria discussed in the following subsection. However, their resource consumption may vary. To identify the most effective encryption scheme, it is essential to evaluate them based on computational efficiency, energy usage, and storage requirements. The appropriate scheme is selected by considering constraints such as data size, processing time, memory, security, and the technology or algorithm used for encryption and decryption. Given the evolving nature of attacks that continuously exploit data through innovative methods, the industry requires new schemes annually to safeguard information. Experimental findings indicate that the enhanced homomorphic encryption scheme meets industry needs, demonstrating improved speed, reduced memory usage, lower power consumption, and IND-CCA security. Consequently, this new EHC is particularly well-suited for applications in wireless sensor networks, bio-healthcare, mobile ad hoc networks, and cloud computing.

4.2.Why this scheme?

1. Accessing an encrypted database hosted on a remote server is essential for data mining and web services.
2. Performing computations on encrypted data is necessary for Ad Hoc Sensor Networks, Wireless Sensor Networks, and cloud-related services.
3. Filtering spam from encrypted emails is relevant for web and cloud services.
4. Outsourcing various types of private computations is primarily applicable in the fields of bio-healthcare and cloud computing services.

5. Security of the Encryption Scheme

Our proposed scheme demonstrates superior security compared to existing methods for several reasons: 1. It employs secret keys q, m, and r, along with a shared key p for encryption, making it extremely challenging to uncover the secret keys. 2. The shared key p is exclusively exchanged between the sender and receiver, complicating the discovery of keys q and r. 3. A random number 'r' is generated for each encryption, ensuring that the same plaintext results in different ciphertexts, thereby making it difficult for adversaries to trace the original plaintext even under close scrutiny. 4. Adversaries cannot access the secret values or the random number. 5. The scheme supports various operations, including addition, multiplication, mixed addition, and mixed multiplication. 6. By utilizing a large prime number p, the decryption process is extended, allowing for a second multiplication. 7. It is an IND-CCA secure scheme, which will be demonstrated in the following section. 8. Additionally, it operates faster than current schemes while consuming less power and memory. 5.4. The non-deterministic feature enhances security, as the random number 'r' ensures that the same plaintext is transformed into different ciphertexts with varying values of r. For example, if p=11, q=7, r=2, x1=5, x2=3, then m=77 results in ciphertext Y1=27 and Y2=25. By altering the random number, the same plaintext can yield different ciphertexts: for p=11, q=7, r=4, x1=5, x2=3, m=77, the ciphertexts become Y1=49 and Y2=47.

5.1. Performance of Our New Homomorphic Encryption Scheme

We executed our algorithm and assessed its execution time. The large integer multiplication and addition were carried out using the GNU Multiple Precision (GMP) Arithmetic Library [10]. We report the milliseconds taken for data encryption and decryption. The computations were conducted on a 2.16GHz Intel Core 2 Duo Processor. The security parameters considered were keys of 512, 1024, and 2048 bits. Data was collected from running 10,000 values for both encryption and decryption, demonstrating the algorithm's high efficiency. For instance, with a key size of 1024 and m=707, the encryption process takes only 10 milliseconds, while decryption takes 8 milliseconds. Given that p is a large prime and q is also prime, we can randomly select r, which corresponds to m chosen plaintexts in an attack that our algorithm can resist. This characteristic renders the algorithm suitable for real-world applications, where large-scale plaintext attacks are not a concern. Since 'r' is generated randomly, the scheme becomes non-deterministic, ensuring that the ciphertext differs with each encryption.

Table.1 the Comparison with other schemes computational cost and memory cost

Scheme	Execution Time (milli sec)	Code size (bytes)	Memory size (bytes)
EHC Our scheme	9	1780	180
MMH	15	2704	221
Elgamal	23	2908	283

5.2.Power consumption

As stated in reference [39], the power consumption P for each arithmetic operation can be determined using the formula P = V.A.T, where V represents voltage, A signifies current, and T indicates execution time. The table illustrates the energy consumption of finite field arithmetic operations measured in nanojoules. It is important to note that, similar to reference [39], the voltage and current values were assumed to be 3V and 1.8 mA, respectively. The power consumption for our Enhanced homomorphic scheme, along with the FHC and MMH schemes, is calculated using the aforementioned formula. The table displays the power consumption for the implementations discussed in this study when these operations are executed.

Table.2.Power consumption of the our scheme EHC with FHC and MMH when operation addition of two ciphers.

Power	Elgamal	MMH	EHC
0	12.22nj	11.33	10.99nj
2	8.33 nj	7.89	6.67 nj
4	6.34 nj	6.56	6.21 nj

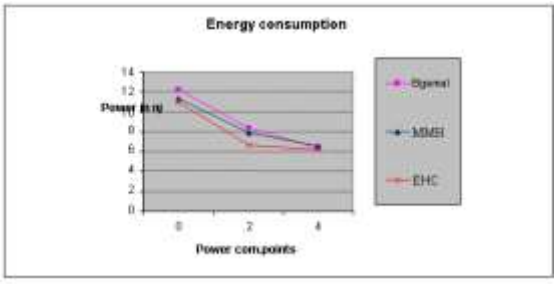


Figure .1 Energy Consumption.

6. Implementation of our EHC in Bio Health Care

This document outlines a novel protocol for comparing private data among parties without disclosing information to one another, utilizing homomorphic encryption techniques. Databases that contain personal, medical, and financial information are classified as sensitive. Typically, an individual's identity is recorded in these databases through their name and personal identification numbers, which are combinations of various attributes. In numerous jurisdictions, it is unlawful to process such data without obtaining a specific license from the relevant authorities. This level of protection is essential to maintain individual privacy and prevent data misuse. A variety of methods have been developed for privacy-preserving queries in distributed database environments, each presenting different trade-offs between security and functionality.

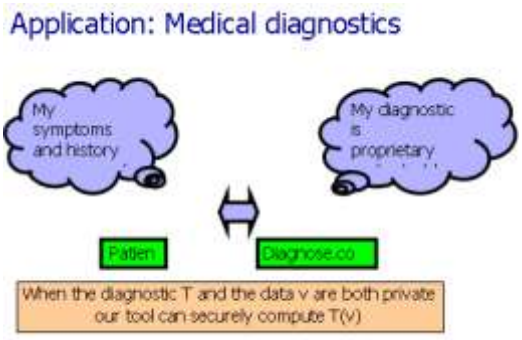
7. Real time application

Recent developments in information processing, along with the capability to exchange and transmit data via flexible and widespread mediums like the Internet and wireless networks, have paved the way for a new type of service. In this model, providers offer their expertise in processing and interpreting data remotely, such as through web services. Examples include the interpretation of medical data for remote diagnoses, access to distant databases, and the processing of personal and multimedia documents. Furthermore, alongside advancements in artificial intelligence, multimedia processing, and data interpretation, as well as affordable access to communication channels, these services necessitate the implementation of security measures to protect user information and the knowledge shared by service providers. Unfortunately, this is often not the case, as data owners typically lack trust in the processing devices or the individuals involved in data manipulation. It is evident that having information processing algorithms capable of operating directly on encrypted data would provide a significant solution to the aforementioned security challenges.

Neural networks (NNs) serve as a cornerstone of contemporary artificial intelligence theory, offering universal capabilities for approximation and generalization, which facilitate a wide range of applications. In this paper, we present a protocol that allows users to request a service provider to execute a neural network using input data in an encrypted format. This protocol aims to achieve two main objectives: firstly, to safeguard the user's data, which serves as the neural network's input, and secondly, to protect the proprietary knowledge embedded in the neural network by the service provider. It is important to clarify that our protocol does not aim to maintain user anonymity. The protection of the network's arc weights and the parameters that define neuron activation functions is central to achieving this goal. Our proposed protocol is based on the principles of homomorphic encryption, which allows for certain basic operations to be performed directly within the encrypted domain. For tasks that cannot be managed...

7.1.Theoretical Analysis

7.1.1.Protocol



Protocol walk through

User encrypts attribute values using an instance of additively homomorphic encryption

Vendor blinds attribute values under encryption for each decision node

Blinded attribute values are converted to Yao circuit inputs using oblivious transfer

Vendor replaces decision nodes with secure integer comparison circuits (offline)

Vendor encrypts each node

User evaluates encrypted branching program

8. Experimental Investigations

The previously mentioned issue can be articulated in real-time as follows. The client possesses an algorithm designed to compute a function f(x). This algorithm is transmitted to the server, which then calculates the function using an input x. This input may be an encrypted version of the client's data stored on the server, necessitating that the function f has the ability to decode it. Importantly, the server must not gain any significant insights into f or the intermediate data utilized during execution. A straightforward protocol for this process is outlined below: 1. The client encrypts f. 2. The client develops a program P(E(f)) that implements E(f). 3. The client forwards P(E(f)) to the server. 4. The server executes P(E(f)) using x as the input. 5. The server returns the result P(E(f))(x) to the client. For instance, Alice, or a doctor, may require a report on their own or their patient's diagnostic data or genomic information to inform future medical recommendations. They would send this data to a database, distributed server, or another doctor (Bob) via the application. The application encrypts and transmits the data to obtain results. On the receiving end, this data is compared with existing information in the databases, and a report is generated without disclosing the data to either party. The application is implemented in C on a Linux platform, and the computational costs and information rates associated with encryption are analyzed, with graphical data provided. The advantages of this approach over existing methods include a reduced reliance on the trustworthiness of the individual parties involved and a heightened focus on privacy concerns related to the sensitive connection between biometric features and their corresponding identities. Notably, this connection remains unknown to both the backend (database) and the frontend (matcher) within the application.

The primary objective is to create a system that allows for the evaluation of encrypted functions without

the need for decryption. These functions can be encrypted in such a way that the resulting transformation can be executed as a program on an unreliable host. The host computer will not have access to the unencrypted instructions of the program but will be able to discern the function's implementation. A significant implication of this is that the executing system will not be able to alter the encrypted function in a goal-oriented manner. Our focus will be on the non-interactive evaluation of encrypted functions, where we encrypt a function while ensuring it remains executable. It is crucial to note that securing only the secret function or data is insufficient; the entire program must be secured. Otherwise, an attacker could modify the plaintext components of the program, causing the secure parts to behave in unintended ways.

Table: 3 Processing time of Enhanced scheme with other Homomorphic schemes

Message size in bits	ElGamal	MMH	EHC encryption scheme
500	42	12	9
1000	42	9	7
2000	42	13	10

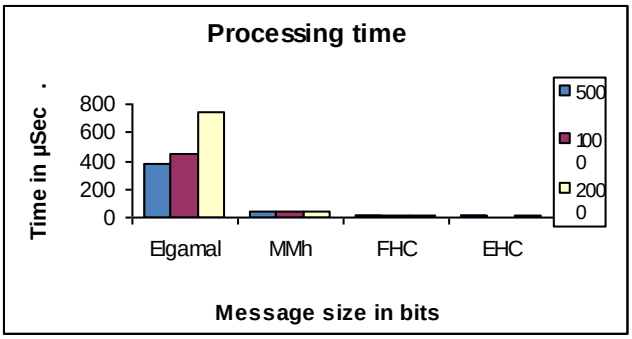


Figure: 2 Processing time of enhanced scheme with other Homomorphic schemes

Figure and respective *Table depicts execution timings of* enhanced scheme with other Homomorphic schemes *in* µSec *with 2048 bit key size.From this* it is clear that EHC is much faster and suitable to this kind of operations in Bioinformatics than MMH and FHC.

9. Conclusion

We have introduced an innovative homomorphic encryption algorithm that does not rely on circuit-based methods. Our approach is fully homomorphic and IND-CCA secure. The security of our encryption scheme is based on the well-established large integer factorization problem, which also underpins RSA, but it necessitates a predetermined limit on plaintext attacks. While Gentry's method and its successors are secure, their time complexity is prohibitively high for practical applications, and their circuit-based design incurs significant overhead. In contrast, our scheme offers a highly efficient time complexity for encryption, decryption, and operations on ciphertexts. Specifically, to counter a chosen plaintext attack involving over 1000 plaintexts, our algorithm performs addition in just a tenth of a millisecond and multiplication in a hundred milliseconds. In comparison, Gentry's algorithm takes over 900 seconds for addition and more than 67000 seconds for multiplication.

Our algorithm's realistic performance and simplicity make it suitable for real-world applications with moderate performance needs, particularly in scenarios where large-scale plaintext attacks are improbable. For instance, in extensive key management systems like certain SCADA systems, utilizing multiple distributed key servers is essential to prevent overload. In such situations, some key servers may not be entirely trustworthy or could be compromised by internal or external threats, leading to severe consequences. Our homomorphic encryption algorithm enables the secure generation, validation, and updating of communication and authentication keys without revealing the actual key information. Additionally, in mobile agent-based systems, agents can employ our homomorphic encryption to safeguard their data and conduct secure transactions on remote hosts. Unlike most existing algorithms that are public-key based, our homomorphic encryption algorithm is symmetric-key based. The primary benefit of public-key homomorphic encryption schemes is the ability to encrypt data without needing access to the private key, allowing multiple clients to make requests to the encrypted database. However, in nearly all applications, it is essential, albeit insecure, for clients to possess the private key to read and decrypt the data in responses. Our request-response communication protocol for the symmetric-key homomorphic encryption scheme enhances security during the request and response processes in multi-user environments. Our innovative method for secure computation, which involves encrypting data into matrices over finite integer rings with eigenvalues corresponding to the plaintext, holds promise for developing even more efficient homomorphic encryption algorithms. We intend to further refine this approach to enhance the efficiency of our scheme, particularly for systems that require protection against chosen plaintext attacks involving numerous plaintext and ciphertext pairs, and to explore additional improvements.

In the field of bio-healthcare, we have developed secure protocols that allow for the comparison of private data without disclosing it to one another through our innovative scheme. Initially, we examined existing homomorphic encryption algorithms, leveraging the homomorphic properties of techniques like Privacycy homomorphism MMH and Fully homomorphic schemes. Subsequently, we introduced a new protocol that enables secure matching of data from database banks or distributed data sources without exposing the data to each other. These new protocols also ensure confidentiality.

9.1.Future Work

The industry is rapidly expanding, leading to a heightened demand for tools and protocols that ensure security in networking. Particularly in critical areas where large volumes of data must be stored, processed, and transmitted to their destinations, implementing this scheme can significantly improve efficiency. As we continue our work, we aim to refine procedures to enhance protocols and technologies in response to market needs. Additionally, we are keen to explore other areas for further enhancements to align with industry demands.

10. References

1. J. Domingo-Ferrer and J. Herrera-Joancomarti. “A privacy homomorphism allowing field operations on encrypted data”. I Jornades de Matematica Discreta i Algorismica, Universitat Politecnica de Catalunya, March 1998.
2. Hyungjick Lee, Jim Alves-Foss,Scott Harrison, “ The use of Encrypted Functions for Mobile Agent Security”,Proceedings of the 37th Hawaii International Conference on System Sciences – 2004.
3. J.Girao, D.Westhoff, and M.Schneider. CDA: Concealed data aggregation for reverse multicast traffic in wireless sensor networks. 40th International conference on communications, IEEE ICC

2005, May 2005.
4. Mithun Acharya, Joao Girao, and Dirk Westhoff. “Secure comparison of encrypted data in wireless sensor networks”. In 3rd Intl. Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, Trentino, Italy, April 2005. WiOpt2005
5. L. Ertaul, ”Cryptography Lecture Notes”, California State University, East Bay, http://www.mcs.csueastbay.edu/~lertaul/
6. Jung Hee Cheon, Hyun Soon Nam,”A Cryptanalysis of the Original Domingo-Ferrer's Algebraic Privacy Homorphism”, http://eprint.iacr.org/2003/221.pdf
7. Yang Xiao. “Security in Sensor Networks”, Auerbach Publications, 2007, pp. 275-290.
8. GSL manual, “http:// www.gnu.org / software /gsl /manual/html_node/Random-Number-Distributions.html”
9. Fork document, “http:/ / www.csl.mtu.edu /cs4411/www/NOTES/process/fork/create. html”
10. GMP manual, “http://gmplib.org / manual/”.
11. Centers for Medicare & Medicaid Services. The Health Insurance Portability and Accountability Act of 1996 (HIPAA). http://www.cms.hhs.gov/ hipaaGenInfo.
12. R. Agrawal, D. Asonov, M. Kantarcioglu, and Y. Li. Sovereign Joins. In *ICDE 2006*, page 26. IEEE Computer Society, 2006.
13. F. Emekc¸i, D. Agrawal, A. E. Abbadi, and A. Gulbeden. Privacy Preserving Query Processing Using Third Parties. In *ICDE 2006*, page 27. IEEE Computer Society, 2006.
14. BRUCE SCHNEIER,”Applied cryptography–Protocols,Algorithms, and Source Code in C “ Second Edition 2009.
15. *Dr. Abu Sayed Md. Latiful Hoque and Gahangir Hossain,”* pir with p-cache: a newprivate information retrieval protocol with improved performance “Malaysian Journal of Computer Science, Vol. 21(1), 2008.
16. Zhiqiang Yang, Sheng Zhong, Rebecca N. Wright,” Towards Privacy Preserving Model Selection “ Preproceedings version, PinKDD’07, August 12, 2007, San Jose, California, USA.
17. Thomas B. Pedersen, Erkay Savas and Yucel Saygin,” secret sharing vs encryption-based techniques for privacy-preserving datamining” Joint UNECE /Eurostat work session on statistical data confidentiality (Manchester, United Kingdom, 17-19 December 2007).
18. C. Negus, "Linux Bible: Boot Up to Fedora, KNOPPIX, Debian, SUSE, Ubuntu, and 7 Other Distributions," 2006..
19. D. Integrity, P. Sakarindr, and N. Ansari, "Security Services IN Group Communications OVER Wireless Infrastructure, Mobile Ad Hoc, AND Wireless Sensor Networks," *IEEE Wireless Communications*, pp. 9, 2007.
20. Alessandro Sorniotti, Laurent Gomez, Konrad Wrona and Lorenzo Odorico “Secure and Trusted in-network Data Processing in Wireless Sensor Networks: a Survey” Journal of Information Assurance and Security 2 (2007) 189 – 199
21. Mufutau Akinwande,” Advances in Homomorphic Cryptosystems” *Journal of Universal Computer Science, vol. 15, no. 3 (2009), 506-522 , 1/2/09 ϟ J.UCS*
22. M. Ilyas, “The Handbook of Ad Hoc Wireless Networks,” CRC Press, 2003.
23. Brett Hemenawy and Rafail Ostrovsky,University of Michigan “On Homomorphic Encryption and Chosen-Cipher text Security “in the Proceedings of PKc 2012.

24. C. Fontaine , F. Galand, A survey of homomorphic encryption for nonspecialists, EURASIP Journal on Information Security, 2007, p.1-15, January 2007
25. D. Micciancio and O. Regev. Post-Quantum Cryptography, chapter Lattice-based Cryptography. Springer, 2008
26. C. Gentry. Fully homomorphic encryption using ideal lattices. In Proc. of STOC, pages 169178. ACM, 2009
27. N. P. Smart and F. Vercauteren. Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes. Lecture Notes in Computer Science, 2010, Volume 6056/2010, 420-443.
28. M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In Advances in Cryptology – Eurocrypt 2010, Springer LNCS 6110, 24–43, 2010.
29. Group Theory by "J S Milne " Version 3.12 April 9, 2012.
30. Brett Hemenway and Rafail Ostrovsky University of Michigan UCLA " On Homomorphic Encryption and Chosen-Ciphertext Security".2012.
31. Jibang Liu, Yung-Hsiang Lu, and Cheng-Kok Koh "Performance Analysis of Arithmetic Operations in Homomorphic Encryption ".2010.
32. Craig Gentry "A FULLY HOMOMORPHIC ENCRYPTION SCHEME ".,2009.
33. Liangliang Xiao, Osbert Bastani, I-Ling Yen ," An Efficient Homomorphic Encryption Protocol for Multi-User Systems ".2011.
34. Gorti V N K V Subba Rao, "Secured data comparison in Bioinformatics using Homomorphic encryption scheme", International Journal Global Journal of Computer Science and Technology ISSN 0975-4172 Sept,2009 Edition **ISSN of Online version:** 0975-4350 72-76 pages.2009.
35. Dr.Allam Appa Rao, Dr.Manga Thayaru,G V N K V Subba Rao,"Analysis of Resistin Protein Involved in Diabetes Associated with obesity using Homomorphic Encryption " **International Journal on Computer Engineering and Information Technology (IJCEIT)** July,2009 Edition.
36. Gorti V N K V Subba Rao, "Privacy homomorphism in Mobile Ad hoc Networks "in an **International Journal of Research and Reviews in Computer Science (IJRRCS)** Vol.2 No.1, P 56, March-2011.
37. Dr.G. Uma, Gorti V N K V Subba Rao, " Role of Homomorphic Encryption Schemes in Wireless Sensor Networks " In an **International Conference CCSB-2009 held at Andhra University-Vizag** on 16-17th Feb,2009.
38. Gorti V N K V Subba Rao , "Application of Homomorphic Encryption Schemes in various fields" in proceedings of a **National Conference at Chaitanya Engg Colleg,vizag - 2008**.
39. Gorti V N K V Subba Rao,Rajasree, " Role of Homomorphic Encryption Schemes in Neural Networks " in proceedings of an **International Conference CCSB-2009 held at Andhra University**-Vizag on 2009.
40. Gorti V N K V Subba Rao , Madan kumar "Discovering the Significance of ICT Exclusively in Education and Research" in **a National Conference ConfER 2008 at VIT,Chennai**.
41. Dr. G. Uma, Gorti V N K V Subba Rao, "Secured data comparison in Bioinformatics using **Homomorphic encryption** scheme", accepted in an International journal of Bioinformatics July 17, 2009.
42. Gorti V N K V Subba Rao, "Secured data comparison in Bioinformatics using **HES** ", accepted in an International Journal of Computational Intelligence in Bioinformatics Aug 26,2009.

43. Dr.G.Uma, Gorti V N K V Subba Rao, "Enhancements in HES " accepted in an **International Journal of Computer Science Issues(IJCSI)** Aug 16,2009.
44. Gorti V N K V Subba Rao, "Enhancements in Homomorphic Encryption Schemes " accepted in International Journal on Computer Engineering and Information Technology (IJCEIT) Nov, 2009 Edition .
45. Vinod Vaikuntanathan University of Toronto " Computing Blindfolded: New Developments in Fully Homomorphic Encryption ".
46. D. Stehl´e and R. Steinfeld, "Faster fully homomorphic encryption," in ASIACRYPT, 2010, pp. 377– 394.
47. David Wagner. Cryptanalysis of an algebraic privacy homomorphism. In Colin Boyd and Wenbo Mao, editors, ISC 2003, volume 2851 of Lecture Notes in Computer Science, pages 234–239. Springer, 2003. 68