



Data Security in Bioinformatics

Gorti VNKV Subba Rao,

Vice Principal,

Sree Dattha Institute Of Engineering And Science –
Hyderabad, India

Mr.A.Yashwanth Reddy,

Assistant Professor,

CSE Dept,Sreedattha Institutions, India

Md.Sameeruddin Khan,

Director,Sree

Dattha Institutions,,
India

Mr.K.Narayana ,

Research Scholar,

Sree Dattha Institutions, India

Abstract: *There are chronic diseases like Diabetics that are associated with considerable morbidity and mortality. It is imperative that more sophisticated, fast, reliable and robust methods need to be devised to develop the best use of information science and technology in relation to decision support and clinical management. In this paper We discussing the new protocol using Homomorphic encryption scheme to compare the input data with the encrypted biometric data in the database banks or distributed data without revealing to each other. We also provide the new scheme that suitable to this type of environment and also the information rate (encrypted) and computational cost of our homomorphic encryption schemes comparing with other scheme.we also show the overall idea implementation and their performanceresults.*

Keywords: *Secure multi-party computation; homomorphic encryption. PH,EHC,MMH.FHC.*

I. INTRODUCTION

Bioinformatics an interdisciplinary field that develops and improves upon methods for storing, retrieving, organizing and analyzing biological data. A major activity in bioinformatics is to develop software tools to generate useful biological knowledge.

Bioinformatics uses many areas of computer science, mathematics and engineering to process biological data. Complex machines are used to read in biological data at a much faster rate than before. Databases and information systems are used to store and organize biological data. Analyzing biological data may involve algorithms in artificial intelligence, soft computing, data mining, image processing, and simulation. The algorithms in turn depend on theoretical foundations such as discrete mathematics, control theory, system theory, information theory, and statistics. Commonly used software tools and technologies in the field include Java, C#, XML, Perl, C, C++, Python, R, SQL, CUDA, MATLAB, and spreadsheet applications. Bioinformatics now entails the creation and advancement of databases, algorithms, computational and statistical techniques, and theory to solve formal and practical problems arising from the management and analysis of biological data. Over the past few decades rapid developments in genomic and other molecular research technologies and developments in information technologies have combined to produce a tremendous amount of information related to molecular biology. Bioinformatics is the name given to these mathematical and computing approaches used to glean understanding of biological processes.

Over the past decade, there has been a growing need for large-scale privacy-preserving systems spanning several databases distributed over the Internet. One motivating example is the nation-wide electronic medical records (EMR) effort within the US which hopes to integrate the EMR of patients across a large number of hospitals while mandating stringent privacy requirements for patient records as specified in the HIPAA regulations [1]. Over the years, the research community has developed a wide range of privacy-preserving techniques for answering different types of queries [2,3,4,13] without revealing information of any individual database which is irrelevant to the queries.

The research also found the following:

- Bioinformatics tools and services have important roles to play in all aspects of drug discovery and development as they help to design drugs, predict drug metabolism and toxicity, and model drug-gene or drug-protein interactions.
- In the post-genomic era, gathering biological information is no longer a bottleneck for scientific researchers. The major hurdle remains in the efficient organization, analysis, and interpretation of the data. The establishment, maintenance and open access of large datasets has been important in driving this field forward, as they have allowed researchers throughout the world to find new ways to analyze and interpret information into new knowledge.
- Raw data is meaningless without context. The ultimate goal of bioinformatics is to extract knowledge from large-scale data. There are currently hundreds of software tools available online, many of which were developed

by leading academic institutions and are freely available, enabling researchers to undertake sequencing, alignment, structure, and function analysis for a range of biological data.

- More data is being collected than can be physically stored; the storage gap is widening, and selecting which data to archive has become a major issue. During the last 30 years, IT infrastructure has become more integrated, and it has rapidly evolved from a computer cluster model to a cloud computing platform that allows computational capacity to be purchased as a service from a cloud computing provider.
- In the Bioinformatics industry, next generation sequencing applications are ideal for cloud migration due to their high performance computing needs as well as the data sharing requirements. However, the savings from cloud are typically lost due to inefficient use of cloud services. A similar trend can be seen with folks attempting to migrate GIS applications to the cloud as well.
- The three key concepts for such applications to really save money in the cloud are **workload partitioning**, **dynamic provider selection**, and **third party security integration**.

Software and tools

Open-source bioinformatics software
Web services in bioinformatics
Bioinformatics workflow management systems
Rosalind

Major research areas

Sequence analysis
Genome annotation
Computational evolutionary biology
Analysis of gene expression
Analysis of regulation
Analysis of protein expression
Analysis of mutations in cancer
Comparative genomics
Network and systems biology
High-throughput image analysis
Structural bioinformatic approaches
Prediction of protein structure
Molecular Interaction
Docking algorithms

II. RACE IS ON IN BIG DATA BIOINFORMATICS MARKET

The race is on for solutions to analyze genetic data faster and easier and to bring the cost down to the point where sequencing becomes accessible for a much wider market.

Bio-IT World reported on Bina's launch of its Genomic Analysis Platform, which helps solve what has quickly become a big data problem in sequencing. The Bina Box is an on-premises hardware and software solution that sits alongside genetic sequencers and captures the files streaming off them, which contain approximately a half-terabyte per sequence. The company said that the box handles assembly and alignment of raw reads and variant calling, which are uploaded to the cloud for comparative analysis, disease-association studies, aggregation, mining, and other downstream analysis. According to an article in *Nature* last week, the market for bioinformatics services and software may soon surpass that for sequencing technologies, as online bioinformatics companies compete to bring genomics platforms and software to hospitals. Even as the cost of sequencing comes down, the infrastructure costs for hospitals and others to analyze the data on their own is still prohibitive.

So, companies are turning to the cloud where they can upload a client's or patients sequencing data and run analysis without the cost of infrastructure. The resulting data can also be shared between doctors or scientists without having to transfer huge files. *Nature* says this will create a new crop of genetics interpretation and analysis firms.

Human Deoxyribo-Nucleic Acid (DNA) sequences offer a wealth of information that reveal among others predisposition to various diseases and paternity relations. the breadth and personalized nature of this information highlights the need for privacy –preserving protocols. The human genome contains a wealth of information about a person's body ; broad access to the genome is likely to revolutionize medical diagnosis and treatment. A doctor can avail this stored data and find out whether the patient has predisposition towards developing a specific diseases like diabetics, blood pressure etc. and also he can understand that the patient reactions towards the specific drug composition. Or whether the treatment will likely fail, there by reducing the overall costs and increasing the effectiveness of the therapy. Finally, it may be possible to create an individual drug therapy for each patient by analyzing his genetic profile and predicting his response to different medications. .If the data is available in internet are with the private organizations then if a patient has digital or image data about his DNA are genome then he wants to give it to another party for diagnosis then obviously both parties wants to maintains secrecy as

Patient feels My symptoms and history are personal Diagnose.com feels My diagnostic is proprietary and valuable

When the diagnostic and the data are both private then we need a procedure to compare both and results will be forward to patient. There are some disadvantages with this kind of data available on internet or with the private databanks

.For instance a person carrying the cancer symptoms may not get insurance coverage which will be rejected at starting itself by insurance companies. In other scenario an employee may be rejected from his permanent job work because of his history from the databanks..

Privacy concerns about this information have traditionally been addressed through laws and procedures .Healthcare professionals are required to keep sensitive data confidential and make it available only with explicit consent of the patient .so far this kind of traditional approach has worked reasonably good ,due to limited data availability at established centers. This kind of traditional form of protecting the sensitive information leakage are insufficient. We may understand that the cryptographic privacy preserving protocols will become invaluable components to over come the procedural approach.

One of the fundamental methods for molecular sequence comparison and alignment is the Needleman-Wunsch algo- rithm [6], which is used in software for detecting similarities between two DNA sequences. The underlying sequence comparison and alignment problem is also known as the string edit problem in the literature. The dynamic programming recurrence relation that subtends the solution to this problem, also serves to solve many other important related problems (either as special cases, or as generalizations that have the same dynamic programming kind of solution). These include the longest common subsequence problem, and the problem of approximate matching between a pattern sequence and text sequence (there is a huge literature of published work for the notion of approximate pattern matching and its connection to the sequence alignment problem). Any solution to the general sequence comparison problem could also be used to solve these related problems. For example, our protocol can remotely compute the answer to the Unix command

```
comp file1 file2 j wc -l
```

where file1 is with Alice and file2 is with Bob, without Alice learning anything about file2 and without Bob learning anything about file1.

III. RELATED SOLUTIONS

The problem discussed here is without revealing the data of the both parties providing the result for further actions to the doctors and diagnosis centers. here we used the homomorphic property to provide security as the computations takes place on encrypted data. Here we are going to discuss some protocols edit distance ,Hamming distance for finding equalities in encrypted data given by some research scholars and also we are providing a protocol using the various homomorphic encryption schemes such as ElGamal, Elliptic curve and privacy homomorphism. They given an efficient protocol for sequence comparisons of the edit-distance kind, such that neither party reveals anything about their private sequence to the other party (other than what can be inferred from the edit distance between their two sequences { which is unavoidable because computing that distance is the purpose of the protocol). The amount of communication done by our protocol is proportional to the time complexity of the best-known algorithm for performing the sequence comparison The problem of determining the similarity between two sequences arises in a large number of applications, particularly in bioinformatics. In these application areas, the edit distance is one of the most widely used notions of sequence similarity: It is the least-cost set of insertions, deletions, and substitutions required to transform one string into the other. The generalizations of edit distance that are solved by the same kind of dynamic programming recurrence relation as the one for edit distance, cover an even wider domain of applications.

In this protocol, Alice has a vector $A = (a_1, \dots, a_n)$ and Bob has a vector $B = (b_1, \dots, b_n)$, where A and B contain only binary values. The two participants are connected together by a secure communication channel. Alice is supposed to learn the Hamming distance of their two vectors, and Alice and Bob are supposed to learn nothing else about each other's vectors. (In the semi-honest setting, such a protocol can easily be transformed into a protocol where both Alice and Bob learn the result, by having Alice tell Bob the answer.) We note that private solutions already exist for this problem. For example, an efficient solution is given by Jagannathan and Wright [12] based on homomorphic encryption. Yao's secure two-party computation could be used [14]; it

computes the result based on computation using a "garbled" circuit. Alternately, the secure two-party computation techniques of Boneh, Goh, and Nissim [6] could be used, in the form where the output is multiple bits; this relies on computationally expensive bilinear pairing, as well as on a new computational assumption. We also note that if one is willing to accept an approximation to the Hamming distance, it is possible to achieve this with sub linear communication complexity and while meeting the privacy requirements [6, 11]. Here they describe a simple, efficient, alternative solution based on the ElGamal cryptosystem. to solve the generalized Hamming distance problem. In this protocol, they assume Alice has an ElGamal key pair (x, y) ($x \in [0, q - 1]$, where q is the size of G ; $y \in G$) such that $y = gx \in G$. Here, x is the private key, which is known only to Alice, and y is the public key, which is also known to Bob. they use $E(m)$ to denote an encryption of m by public key y . All computations in the protocol and throughout this paper take place in G , which is chosen large enough to ensure that the final distance result is correct as an integer. The output of this protocol is the Hamming distance $\text{dist}(A,B) = \sum_{i=1}^n (a_i \oplus b_i)$.

Multiparty users through servers/distributed servers to get the information. a Secure Multi-party Computation (SMC) problem deals with computing any function on any input, in a distributed network where each participant holds

one of the inputs, while ensuring that no more information is revealed to a participant in the computation than can be inferred from that participant's input and output. The SMC problem literature was introduced by Yao [17]. It has been proved that for any polynomial function, there is a secure multiparty computation solution [17]. The approach used is as follows: the function F to be computed is firstly represented as a combinatorial circuit, and then the parties run a short protocol for every gate in the circuit. Every participant gets corresponding shares of the input wires and the output wires for every gate. This approach, though appealing in its generality and simplicity, is highly impractical for large datasets.

One recent trend is to use secure coprocessor (SC) to realize the functionalities provided by SMC. In the work by Agrawal et al. [13], privacy-preserving join operation is done with the help of a SC. Semantic secure (probabilistic) encryption is done by the database on the data value. The SC decrypts them and performs the matching. With the original data recovered, the matching thus can be based on any arbitrary function other than a simple equality check or computation of degree-2 polynomials, in contrast with the many other approaches. On medical data, nodes may not wish to participate in a P2P network to exchange encoded data streams with other unknown nodes.

They observed a Weaknesses that SC is used to realize the assumption of faithful computation and the ideal functionalities of computing a function with the inputs and any intermediate values kept private from anyone. However, from a computational standpoint, it is much more expensive to implement cryptographic operations (e.g. decryption) in the SC in comparison to simple database indexing operations: for decrypting millions of records, the overall system performance will significantly reduce. In addition, the cost of tamper-proof memory also imposes a practical limitation on the type of algorithm that can be performed by the SC. Mobile agents interacting with the server to get the comparisons result about the diseases. In one paper, Sander and Tschudin focus on extending the mobile cryptography approach [13, 14, 15], in terms of privacy and integrity, and explore its usefulness and effectiveness in protecting mobile agents. To extend mobile cryptography, we will consider composite functions and additive-multiplicative homomorphism to encrypt mobile agents. As the contribution of this research, the encrypted mobile agent will be able to run on any host without decryption. The encrypted mobile agent will generate encrypted results, which will be decrypted by the agent owner. This will improve the overall security of the mobile agents. is destroyed before the agent migrates to the next host. Destroying secret keys before agent migration ensures that the previous partial results are secure and intact. Since the agent originator maintains the secret keys, the partial results can be verified on the originator's home site.

IV. HOMOMORPHIC ENCRYPTION

A powerful tool in computing a wide range of functions with computational security is *homomorphic encryption*. A homomorphic encryption function allows manipulation of two (or more) ciphertext to produce a new ciphertext corresponding to some arithmetic function of the two respective plaintexts, without having any information about the plaintext or the encryption/decryption keys. For example, if $E()$ is multiplicatively homomorphic, given two ciphertext $E(A)$ and $E(B)$, it is easy to compute $E(A \cdot B)$. Whereas, if $E()$ is additively homomorphic, then computing $E(A + B)$ is also easy. One well-known example of a multiplicatively homomorphic encryption function is textbook RSA. An example of an additively homomorphic encryption function is Paillier [14]. In more detail (as described in [13]) a homomorphic encryption function can be defined as follows: Homomorphic encryption functions were originally proposed as a method for performing arithmetic computations over private databanks [12]. Since then, they have become part of various secure computation schemes and more recently, homomorphic properties have been utilized by numerous digital signature schemes [13, 4]. As mentioned above, some encryption functions are either additively or multiplicatively homomorphic. An open problem in the research community is whether there are any cryptographically secure encryption functions that are both additively and multiplicatively homomorphic. (It is widely believed that none exist.)

In their paper, the problem of decrypting data before applying arithmetic operations is addressed and a new approach is described as processing data without decrypting it first. Succeeding works showed that some asymmetric cryptosystems preserve structure, which allows for arithmetic operations to be performed on encrypted data. This structure preserving property, called homomorphism, comes in two main types, namely, additive and multiplicative homomorphism. Using additive homomorphic cryptosystems, performing a particular operation (e.g., multiplication) with encrypted data, results in the addition of the plaintexts. Similarly, using a multiplicatively homomorphic cryptosystem, multiplying ciphertext, results in the multiplication of the plaintexts. Paillier [14], Okamoto-Uchiyama [18], and Goldwasser-Micali [12] are additively homomorphic cryptosystems while RSA [13] and ElGamal [14] are multiplicatively homomorphic cryptosystems. Computing will be done on encrypted data by implementing the homomorphic encryption scheme. So that computing will be done on encrypted functions and encrypted data. The encryption scheme should also be public-key and semantically secure, i.e., $E(a)$ gives no information about a . Several steps of our protocol use homomorphic encryption.

V. NEW SCHEME: ENHANCED HOMOMORPHIC ENCRYPTION SCHEME (EHES)

We proposed a new Enhanced Homomorphic Cryptosystem (EHC) for homomorphic Encryption / Decryption with IND-CCA secure scheme [5] exhibit better performance than existing schemes mainly in processing speed, memory and power consumption. Our scheme is non deterministic and exhibits addition, multiplication, mixed addition and mixed multiplication operations.

Our Construction. A large prime number ' p ', another prime number ' q ' such that $q < p$ are taken and a random number ' r ' has taken to make the scheme non deterministic. Consider the set of clear text data Z_p and the set of clear

text operations {+, -, *, / and mixed} consisting respectively, of the addition, subtraction, multiplication and mixed multiplication modulo m, with $m = pq$. Let the cipher text data set be Z_c . Define the encryption key $k = (p, q, m, r)$ and $E_k(X) = Y = (X + r * p^q) \pmod{m}$. Decryption will be done with the secret key $k = (p)$, $X = D_k(Y) = C \pmod{p}$. But can be broken if p can be discovered which is not easy to solve.

A computer can factor that number fairly quickly, but (although there are some tricks) it basically does it by trying most of the possible combinations. One can find two huge prime numbers, p and q that have 200 or may be 400 digits each. Q will be kept secret (It is secret key), and by multiplying them together to make a number $m = pq$. That number m is also a secret key to encrypt the data.

It is relatively easy to get m by multiplying p and q . But if anybody know m , it is basically impossible to find p and q . To get them, you need to factor m , which seems to be an incredibly difficult problem finding the 'r' also difficult as this value will be generated randomly. It is generally regarded that m should be at least 1024, if not 2048.

Let us see example to understand the cipher and decipher procedure.

Proof : Encrypt the message X

$$E(X) = Y = (X + r * p^q) \pmod{m}$$

Cipher text Y will be $(X+rp)$

Decrypt $Y = X = Y \pmod{p}$

$$= (X+rp) \pmod{p}$$

$$= rp \pmod{p} + X \pmod{p}$$

$$= X \text{ Plaintext}$$

VI. SECURITY OF THE ENCRYPTION SCHEME

We can say our scheme has all the features of mentioned bellow and is more secure when compare to existing schemes as follows:

1. It uses the secret keys $q, m,$ and r and sharing key p for encryption. So it takes time to find the secret keys.
2. the shared key p only shared between the sender and receiver which is not simple to find the q and r .
3. Random number 'r' will be generated randomly so that every time the same plaintext mapped to different cipher text so that it to track the plaintext even with strong observation for opponent.
4. Opponent can not get the secret value and random number.
5. EHES supports Addition, Multiplication, Mixed addition and Mixed multiplication.
6. As we are also taking large prime number p the decryption circle will be more so that second multiplication also possible.
7. EHES is IND-CCA secured scheme which will be proved in the next section.
8. It's performance is relatively better than the existing schemes defined in [19] and consumes less power and memory.

PROPERTIES OF THE EHES

- As p is very large so that the decryption circle is more that allows second multiplication operation on encrypted data.
- q is also a prime and kept secret along with m increases the security.
- r is taken randomly that makes the scheme non deterministic.

The random number 'r' provides the feature non deterministic which means the plaintext will be converted into different ciphertext with the change in the value r . We can better understand using the following example.

Let $p=11$ $q=7$ $r=2$ $x_1=5$ $x_2=3$ then $m=77$

cipher text $Y_1= 27$ cipher text $Y_2= 25$

Now by changing the random number the same plain text will be mapped to another cipher texts

Let $p=11$ $q=7$ $r=4$ $x_1=5$ $x_2=3$ then $m=77$

cipher text $Y_1= 49$ cipher text $Y_2= 47$

- As the conditions and validations are less that reduces the processing time.
- Memory required also less.
- Decryption process takes less time.
- Knowledge of ciphertext or any other value will not reveal the p .
- It supports all the arithmetic operations and the overhead also reduced caused by circuit based environment as in Gentry's scheme.
- Second multiplication also allowed $D(E(x)*E(x)*E(x)) = x*x*x$ without revealing the x .

VII. PERFORMANCE OF ENHANCED HOMOMORPHIC ENCRYPTION SCHEME

We implement our algorithm and evaluate its execution time. The large integer multiplication and addition were implemented using the GNU Multiple Precision (GMP) Arithmetic Library [29]. We give the number of milliseconds required to perform encryption and decryption of data. The computations were performed on a 2.16GHz Intel Core 2

Duo Processor. The security parameter considered was key = 512, 1024 and 2048. The data was gathered from running 10000 values. For encryption and decryption the algorithm is efficient. For example, for key = 1024 and for m=707 the algorithm runs encryption in only 10 milliseconds and runs decryption in 8 millisecond. For such p is a very large prime and q also a prime, we can choose r randomly, which translates into m chosen plaintexts in an attack that our algorithm can withstand. This makes the algorithm practical for real world implementation where large scale plaintext attacks are not an issue. As the 'r' will be generated randomly the scheme become non deterministic so that the cipher text will be different for every encryption.

Name Of The Scheme	Additive	Multiplicative	Mixed Addition And Multiplicative
Elgamal	NO	YES	NO
FHC	YES	YES	YES
ECC	YES	NO	NO
MMH	YES	YES	YES
EHES	YES	YES	YES

EXISTING HOMOMORPHIC ENCRYPTION SCHEMES.

For the purpose of comparison, we estimated the computation time of Gentry's homomorphic encryption scheme [8] and Mixed multiplication scheme. In [77], the performance of the primitive operations has been studied: The bootstrapping (recrypt) time is 6 seconds, which dominates the time of the operations. For 32-bit numbers, the addition circuit needs 5 gates and the multiplication circuit needs 112 gates [15]. Thus, Gentry's homomorphic encryption scheme needs $5 \cdot 32 \cdot 6 (> 900)$ seconds to add two 32 bit numbers, and $11 \cdot 32 \cdot 6 (> 67000)$ seconds to multiply two 32 bit numbers. This is far slower than our homomorphic encryption scheme[17].

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

Performance comparison of EHES with MMH and FHC.

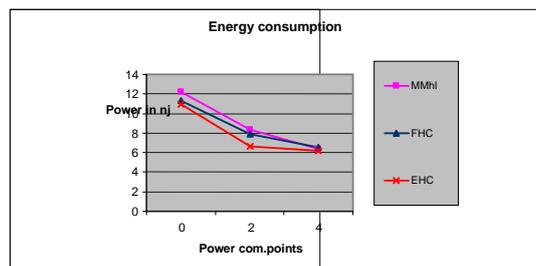
Power consumption

According to [34], the power consumption P of each arithmetic operation can be calculated by using the formula $P = V \cdot A \cdot T$ whereby V denotes the voltage, A denotes the current, while the execution time is represented by T. Table represents the energy consumption of the finite field arithmetic operations in nanojoules. Note that similar to [39], the voltage and the current was assumed to be 3V and 1.8 mA, respectively.

The power consumption of our Enhanced homomorphic scheme with FHC and MMH scheme are calculated by the formula mentioned above. Table represents the power consumption of the implementations from this work, when those operations are performed.

Power com.points	MMH	FHC	EHC
0	12.22nj	11.33 nj	10.99nj
2	8.33 nj	7.89 nj	6.67 nj
4	6.34 nj	6.56 nj	6.21 nj

Power consumption of EHES under addition of two ciphers.



Power consumption.

Security attacks

- Ciphertext-only attack
- Known-plaintext attack
- Chosen-plaintext attack
- Chosen-ciphertext attack
- Adaptive chosen-ciphertext attack
- Indistinguishable chosen cipher text (IND-CCA) attack

Chosen cipher text security (IND-CCA) considered as the right notion of security for a cryptosystem.

VIII. OVERALL IDEA OF THIS APPLICATION

Alice or doctor wants report of his or his patient diagnosis data/image of his genome or DNA for understanding about his future medical suggestions so he will send the data to the database bank or distributed server or Bob/another doctor through the application .the application itself encrypts and sends data for result .At other side this data will be compared with the available data in banks and sends the report without revealing the data to each other. the application is simulated in c under Linux platform and the computational costs and information rates in encryption are compared and the graphical information has been furnished here. Cryptographic protocols using homomorphic encryption may also allow us to compare directly encrypted data. For instance, Schoenmakers and Tuyls improve Paillier’s public encryption protocol and propose to use it for biometric authentication protocols by employing multi-party computation techniques [12]. we assume that the biometric information is public, but the relationship between a user’s identity and its biometric information is private. The advantages over the existing ones: The first is that we lower the level of trust on the involved individual principals. The second is that extra attention has been paid to the privacy issues related to the sensitive relationship between a biometric feature and the relevant identities. Specifically, this relationship is unknown to the database and the matcher.

The central idea is to develop a system on which we can evaluate encrypted functions without decrypting it. The functions can be encrypted such that the resulting transformation can be implemented as a program that can be executed on an unreliable host . The executing computer will not be able to see the program's unencrypted instructions but will not be able to make out what the function implements. The important implication here is that the executing system won’t be able to modify the encrypted function in goal – oriented way. We will be looking at non interactive evaluation of encrypted functions , where in we encrypt a function such that it still remains executable. The first observation we can make is that it is not enough to secure only the secret function or secret data, the whole program has to be made secure otherwise an attacker can modify the clear text parts of the program and make the secure parts do something other than what it is meant to do.

Experiment Procedure and Results

The problem can be described as follows

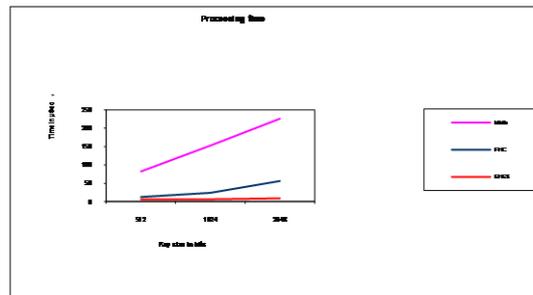
The client has an algorithm to compute a function $f(x)$. The client sends it over to the server which computes the function on an input x . x can be encrypted form of client's data stored by the server. Therefore the function f should have the capability to decode it . The server should not be able to learn anything substantial about f or the intermediate data it uses for execution.

In our simulation we have measured and determined the encryption timings of all above mentioned encryption schemes by varying the key size (512, 1024, 2048 bits) and keeping the message size fixed (512 bits). In another run we determined the execution timings of all these same encryption schemes by keeping the key size fixed (512 bits, 1024 bits , 2048 bits) and varying message size. The timings are determined over 200 runs.

Figure 6.2 represents the execution timings of Table 6.1, by observation we found clearly that EHES is better than other encryption schemes . We also observed that the encryption timings of Scheme MMH and FHC increases with the increase in encryption keys but in case of EHES the encryption timing remains almost the same with the increase in the encryption key size.

	MMH	FHC	EHES
512	83	13	7
1024	153	24	7
2048	226	56	9

Execution timings of schemes in micro seconds by varying the key size (512, 1024, 2048 bits) and keeping the message size fixed (512 bits)



Processing time of Schemes in micro seconds with varying key sizes and fixed message size 512 bits.

Above Table represents the execution timing of above mentioned four schemes in micro seconds by increasing the message size to 100, 250 and 500 bits and by keeping the key size fixed (512 bits). Figure 6.3 graph represents the execution timings of Table From this Figure it is very clear that EHEC is much better than other two schemes. We also found that the encryption timing of other schemes increases with the increase in message size we also observed that the encryption timings of EHEC remains almost the same with the increase in the message size.

IX. CONCLUSION

Hence, the most important points that make our more appropriate for biometrics authentication protocols are the following. Firstly, no secret information storage is required at the client side. Secondly, the protocol guarantees the privacy of the relationship between the user's identity and its biometric data, and the privacy of the user's biometric information. we considered a biometric authentication protocol where confidentiality is required for biometric data solely for privacy reasons. We captured these notions into a security model and introduced a protocol which is proved secure in this security model. It remains an interesting issue to improve its performance. Our new scheme is suitable to this environment.

REFERENCES

- [1] Centers for Medicare & Medicaid Services. The Health Insurance Portability and Accountability Act of 1996 (HIPAA). <http://www.cms.hhs.gov/hipaaGenInfo>.
- [2] R. Agrawal, D. Asonov, M. Kantarcioglu, and Y. Li. Sovereign Joins. In *ICDE 2006*, page 26. IEEE Computer Society, 2006.
- [3] F. Emekci, D. Agrawal, A. E. Abbadi, and A. Gulbeden. Privacy Preserving Query Processing Using Third Parties. In *ICDE 2006*, page 27. IEEE Computer Society, 2006.
- [4] M. Naor, B. Pinkas, and R. Sumner. Privacy Preserving Auctions and Mechanism Design. In *Electronic Commerce 1999*, pages 129–139. ACM, 1999.
- [5] A. C. Yao. Protocols for secure computations. In *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science*, 1982.
- [6] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.
- [7] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1986.
- [8] T. Sander and C. Tschudin. Towards mobile cryptography. Technical report, International Computer Science Institute, Berkeley, 1997.
- [9] T. Sander and C. Tschudin. On software protection via function hiding. In *Information Hiding*, pages 111–123, 1998.
- [10] T. Sander and C. Tschudin. Protecting Mobile Agents Against Malicious Hosts. In G. Vigna, editor, *Mobile Agent Security*, pages 44–60. Springer-Verlag: Heidelberg, Germany, 1998.
- [11] H. Lee. *Mobile Agent: Evaluating Encrypted Functions*. PhD thesis, Department of Computer Science, University of Idaho, August 2002.
- [12] D. Naccache and J. Stern. A new public-key cryptosystem. In *Theory and Application of Cryptographic Techniques*, pages 27–36, 1997.
- [13] Y. Sakurai, M. Yokoo, and K. Kamei. An efficient approximate algorithm for winner determination in combinatorial auctions. In *Proceedings of the Second ACM Conference on Electronic Commerce (EC-00)*, pages 30–37, 2000.
- [14] T. Sandholm. An algorithm for optimal winner determination in combinatorial auction. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 542–547, 1999.
- [15] M. H. Rothkopf, A. Pekec, and R. M. Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44(8):1131–1147, 1998.

- [16] Y. Fujishima, K. Leyton-Brown, and Y. Shoham. Taming the computation complexity of combinatorial auctions: Optimal and approximate approaches. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 548–553, 1999.
- [17] M. Tennenholtz. Some tractable combinatorial auctions. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, pages 98–103, 2000.
- [18] R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.
- [19] S. B. Needleman and C.D. Wunsch. A General Method Applicable to the Search for Similarities in the Amino-acid Sequence of Two Proteins, *Journal of Molecular Biology* 48, pp.443{453 (1973).
- [20]. B. Schoenmakers and P. Tuyls. Efficient binary conversion for Paillier encrypted values. In S. Vaudenay, editor, EUROCRYPT, volume 4004 of Lecture Notes in Computer Science, pages 522–537. Springer, 2006.
- [21] Brett Hemenawy and Rafail Ostrovsky, University of Michigan “On Homomorphic Encryption and Chosen-Cipher text Security” in the Proceedings of PKC 2012.