# IMPLEMENTATION OF HOMOMORPHIC ENCRYPTION SCHEME TO SAFEGUARD CLIENT DETAILS

**[1]P.RANJIT JEBA THANGAIAH, [2]T.ARUMUGA MARIA DEVI, [3]M. ANGEL ISWARYA**

[1]Assistant Professor, Karunya University, Coimbatore., [2]Assistant Professor
[3]Research Scholar, Centre for Information Technology & Engineering, Manonmaniam Sundaranar University, Tirunelveli, Tamil Nadu

**Abstract**- The demand for privacy of digital data and of more complex structures like an algorithm has become stronger during the last few years. The privacy of personal information on the Internet has become a major concern for governments, businesses, media, and the public. On the one hand, this personal information enables a great variety of attacks on digital goods and on the other hand they are vulnerable to attacks such as the manipulation or destruction of data and the theft of sensitive information. For storing and reading data securely, this paper covers an encryption scheme is "homomorphic" used for the data security. In order to prevent the leakage of information from IT companies, this paper uses homomorphism algorithm which has data encryption technique.

## I. INTRODUCTION

1.1 Overview

With growing data, secure communication and data security is of paramount importance. Today one way to achieve secure communication is by the use of cryptography, which concurrently ensures confidentiality of data in communication and in storage. The focus has been mostly on how to protect the individual from being tracked. The main objective of this paper, to provide data security through Homomorphism algorithm.

For a secured system it has been desired to make the combination of data and keys secured. For storing and accessing data securely there exist many ways which can guarantee privacy and confidentiality.

Secure privacy Homomorphic could be applied to protect database against eavesdropping. Homomorphic cryptography provides a third party with the ability to perform simple computations on encrypted data without revealing any information about the data itself. Homomorphic encryption improves the efficiency of secure multiparty computation.

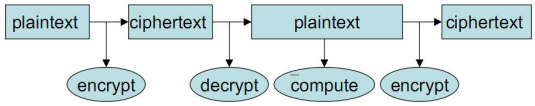
In order to avoid the leakage of sensitive data, the data stored in the database in the encrypted format with the help of Homomorphism Algorithm. It is very helpful to maintain confidentiality, integrity, and provides a high security of data. In this paper, Homomorphism algorithm is implemented for IT concern to secure the sensitive data.

In order to prevent the leakage of information from IT companies, this system uses homomorphism algorithm which has data encryption technique. This homomorphism algorithm provides high level of security to preserve privacy. This method can make the sensitive data to be appeared with secret content and prevent from misusing. In this paper, we implement the homomorphism algorithm, holds the details of customers and project work status such as address, contact numbers, mail id, and current status of the project etc., stored in the database in the encrypted format.

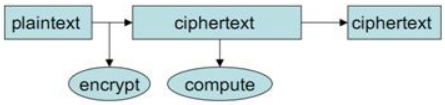
This system is designed to allow the authenticated people can only access and retrieve data. Except browser there is no special hardware, software devices are needed to implement the Homomorphism algorithm.

## II. BASIC CONCEPTS

Ordinary encryption can't compute the ciphertext data




Homomorphic Encryption scheme (HES) can do it and furthermore encrypt operation value automatically.




2.1 Homomorphic encryption

In this section the basic definitions related to homomorphic encryption. Let M (or C) denote the set of the plaintexts (or ciphertexts, respectively). An encryption scheme is said to be homomorphic if for any given encryption key k the encryption function E satisfies

$$\forall m1,m2 \in M, E(m1 \bullet_M m2)=E(m1)\bullet_C E(m2) \quad (2.1)$$

For some operators $\bullet_M$ in M and $\bullet_C$ in C, where = means "can be directly computed from," that is, without any intermediate decryption [Fontaine and Galand, 07].

Informally speaking, homomorphic cryptosystem is a cryptosystem with the additional property that there exists an efficient algorithm to compute an encryption of the sum or the product, of two messages given the public key and the encryptions of the messages but not the messages themselves.

If M (or C) is an additive (semi-) group then the scheme is called additively homomorphic and the algorithm is called Add. Otherwise the scheme is called multiplicatively homomorphic and the algorithm is called Mult.

A. Abbreviations and Acronyms

General Terms
Data Security.

Keywords
Homomorphic Encryption, Crypto system, Sensitive data, privacy. Plain text, cipher text

B. Privacy Homomorphism

1. D and E are easy to compute.
2. The functions fi' and predicates pi ' in C are efficiently computable.
3. E is a non-expanding cipher or an expanding cipher whose cryptotext has a representation only marginally larger than the corresponding plaintext.
4. The operations and predicates in C should not be sufficient to yield an efficient computation of D.

Homomorphic Encryption scheme(HES)

- Encryption:y = E(x) = a mod n, where a = x + rp (2.2)
- Decryption:x=D(y) = y mod p (2.3)

Additively Homomorphic

1.y=E(x)=smod((x+sign(x)×rand()×p),n). (2.4)
rand() yields a random plus integer, sign(x) produces a plus or minus symbol like x
2.Given x1 and x2, to compute x1 + x2, first of all calculating y1= E(x1) and y2= E(x2) according to 1.
3.If | x1 |-| x2 |≥0 ,then | y1 |-| y2|≥0, otherwise return 2 to recalculate y1 or y2. as far as the condition is satisfied.

Calculating y1+ y2, the result is encrypted automatically. Decryption Given y = E(x), use the key p to recover x = D(y) = smod(y,p).

## III. SYSTEM FEATURES

The modularization approach allows for more robust application with fault tolerance and easy module replacement for security and Upgradeability.

3.1 Login
The project is accessed by the employee of the IT sectors by entering a valid username and password. New employee enter into the organization admin allot the username and password for that employee. This phase has highest priority as it handles the authentication.

3.2 Employee Information module
This module handles the employee personal, official details. The official details are divided into the following sub modules.

- Department
- Educational Background
- Experience

**3.3** Client Information Encryption module
This module handles the details about the client. In order to secure the client details, homomorphism encryption algorithm is used. The client details are encrypted by the secret code using Homomorphism Encryption algorithm. Authorized user can only know about that code. The encrypted data is stored into the database. Intruder cannot understand the data.

3.4 Client Information Decryption module
This module handles the users can view decrypted or original data using the secret code, which is given by user during the encryption time. During the decryption process encrypted data are taken from database as input and the original data are getting as output using the Homomorphism algorithm.

3.5 Project details
This module handles the complete details about the project like name of the project, duration of the project, hardware, software specification, client id, status of the project and description. And also the project is divided into modules and forms. Maintain the details about the head of developer, tester. The sub modules are

- Project Work Status
- Project Status Update

3.6 Bugs
This module allows the tester to post the bugs. The sub modules are

- Bug recheck
- Bug Track.

3.7 Feedback
This module determines the client can enter the feedback about the project.

3.8 Administration module
The administrator handles

- User Management
- Authentication
- Client Management
- Department Management

- Project Management
- Feedback.
- Report Generation

## IV. EXPERIMENTAL SETUP

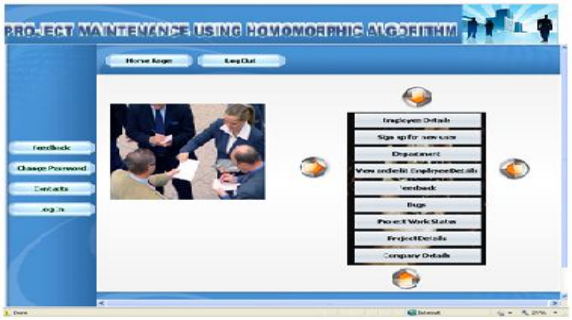



**Figure 4.1 Different types of user developer, tester, admin login to the system.**

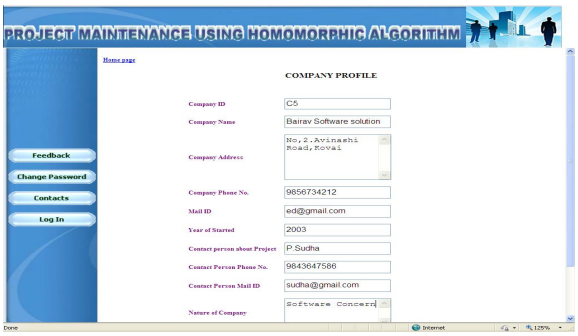



**Figure 4.2 Encrypt the client details**

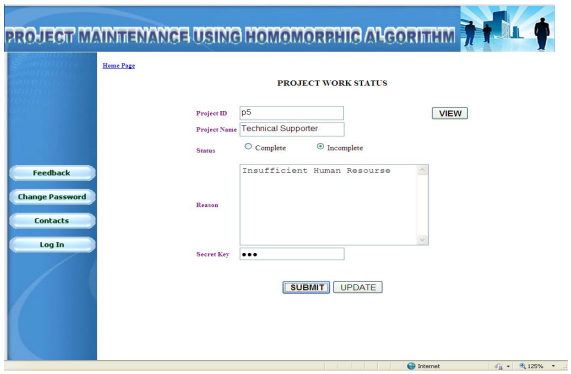



**Figure 4.3Encrypt project work status**

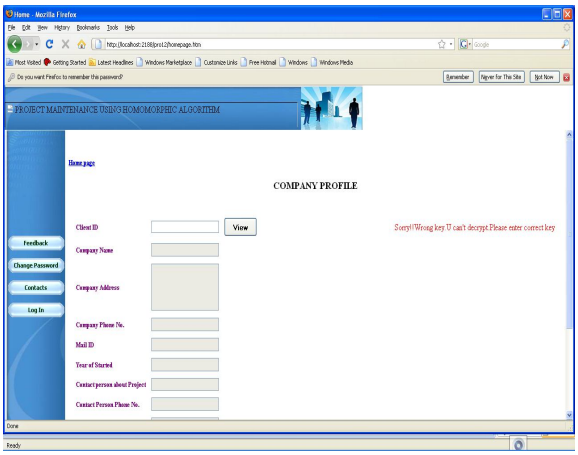



**Figure 4.4 Decrypt Client Details**

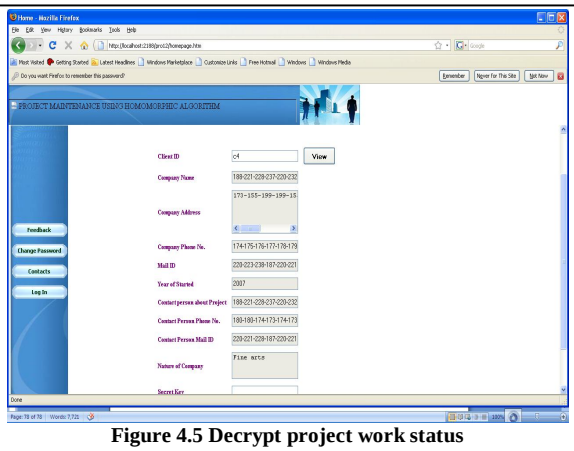



**Figure 4.5 Decrypt project work status**

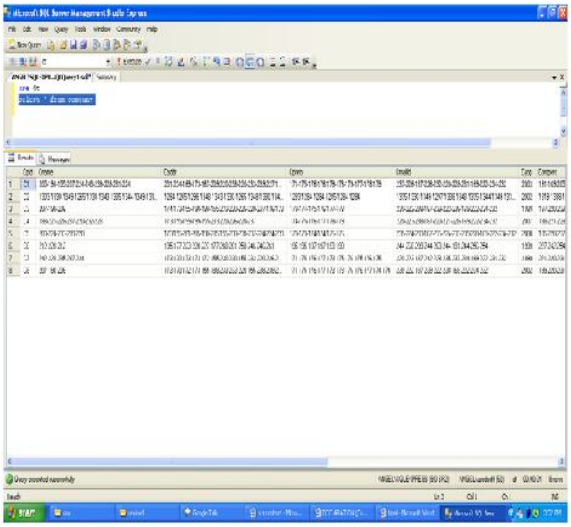


**Figure 4.6 Client details are stored in the data base in the encrypted format using homomorphism algorithm.**

## CONCLUSION

Thus the implementation of Homomorphic encryption scheme, which allows the user to secure the sensitive data. To avoid information exposing of personal privacy, personal data was used as the encrypted principle, applying homomorphism to strengthen personal data security. Prevent personal data from being misused. Applying homomorphism for IT concern, protect the client details and the project work status. This system is designed to allow the authenticated people can only access and retrieve data.

## ACKNOWLEDGEMENT

First of all, we wish to offer my sincere prayers and thanks to the god almighty for his unfailing grace and miraculous blessing towards us throughout this paper. I wish to express my love and respect for my husband, parents for their support, sacrifice, contribution and encouragement, which helped me a lot to complete this paper successfully. I express my sincere and humble thanks to Dr.T.Arumuga Maria Devi, supervisor for her endless support, patience and encouragement towards the work.

## REFERENCES


[1] M. V. Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, Fully Homomorphic Encryption over the Integers. EUROCRYPT 2010: pp. 24-43, June 2010. [2]Bruce Schneier (2009-07-09). "Schneier on Security: Homomorphic Encryption".

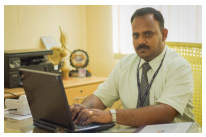
[2] Craig Gentry. Fully Homomorphic Encryption Using Ideal Lattices. In the 41st ACM Symposium on Theory of Computing (STOC), 2009.

[3] D. Wagner, "Cryptanalysis of an algebraic privacy homomorphism," in Proceedings of the 5th Inter- national Conference on Information Security (ISC 2003), pp. 234-239, Springer-Verlag, 2003.

[4] Nigel P. Smart and Frederik Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In Phong Q. Nguyen and David Pointcheval, editors, Public Key Cryptography, volume 6056 of Lecture Notes in Computer Science, pages420-443. Springer, 2010.

[5] R. Rivest, L. Adleman, and M. Dertouzos. On data banks and privacy homomorphisms. In Foundations of Secure Computation.


## AUTHORS



P.Ranjit Jeba Thangaiah received Bachelor degree in physics from P.S.G college of Arts and Science,Coimbatore,India.Received M.C.A. degree from Karunya Institute of Technology.Received his Ph.D degree from Bharathiar university, coimbatore.His research interest include Data mining, Machine learning.

T. Arumuga Maria Devi Received B.E. Degree in Electronics & Communication Engineering from in Manonmaniam Sundaranar University, Tirunelveli India in 2003, M.Tech degree in Computer & Information Technology from Manonmaniam Sundaranar University, Tirunelveli, India in 2005 and also Worked as Lecturer in department of Electronics & Communication Engineering Sardar Raja College of Engineering. Received Ph.D Degree in Information Technology.

M. Angel Iswarya received M.C.A degree from Karunya University, Coimbatore, India in 2010.Received PG diploma in Cyber Security from Annamalai University, Chidambaram, India in 2011. Currently she is doing research in Information Technology & Engineering from M.S Universiy, Tirunelveli,India. Her Research interests include Data Communication and Security in Computing.

★★★