ABSTRACT:

An algorithm for anonymous sharing of private data among parties is developed. This technique is used iteratively to assign these nodes ID numbers ranging from 1 to $N$. This assignment is anonymous in that the identities received are unknown to the other members of the group. Resistance to collusion among other members is verified in an information theoretic sense when private communication channels are used. This assignment of serial numbers allows more complex data to be shared and has applications to other problems in privacy preserving data mining, collision avoidance in communications and distributed database access. The required computations are distributed without using a trusted central authority. Existing and new algorithms for assigning anonymous IDs are examined with respect to trade-offs between communication and computational requirements. The new algorithms are built on top of a secure sum data mining operation using Newton’s identities and Sturm’s theorem. An algorithm for distributed solution of certain polynomials over finite fields enhances the scalability of the algorithms. Markov chain representations are used to find statistics on the number of iterations required, and computer algebra gives closed form results for the completion rates.
EXISTING SYSTEM:

A secure computation function widely used in the literature is secure sum that allows parties to compute the sum of their individual inputs without disclosing the inputs to one another. This function is popular in data mining applications and also helps characterize the complexities of the secure multiparty computation.

DISADVANTAGES OF EXISTING SYSTEM:

The algorithms for mental poker are more complex and utilize cryptographic methods as players must, in general, be able to prove that they held the winning hand. Throughout this paper, we assume that the participants are semi-honest, also known as passive or honest-but-curious, and execute their required protocols faithfully. Given a semi-honest, reliable, and trusted third party, a permutation can also be created using an anonymous routing protocol.

PROPOSED SYSTEM:

This work deals with efficient algorithms for assigning identifiers (IDs) to the nodes of a network in such a way that the IDs are anonymous using a distributed computation with no central authority. Given N nodes, this assignment is
essentially a permutation of the integers \{1,\ldots,N\} with each ID being known only to the node to which it is assigned. Our main algorithm is based on a method for anonymously sharing simple data and results in methods for efficient sharing of complex data.

Despite the differences cited, the reader should consult and consider the alternative algorithms mentioned above before implementing the algorithms in this paper. This paper builds an algorithm for sharing simple integer data on top of secure sum. The sharing algorithm will be used at each iteration of the algorithm for anonymous ID assignment (AIDA). This AIDA algorithm, and the variants that we discuss, can require a variable and unbounded number of iterations.

The work reported in this paper further explores the connection between sharing secrets in an anonymous manner, distributed secure multiparty computation and anonymous ID assignment. The use of the term “anonymous” here differs from its meaning in research dealing with symmetry breaking and leader election in anonymous networks. Our network is not anonymous and the participants are identifiable in that they are known to and can be addressed by the others. Methods for assigning and using sets of pseudonyms have been developed for anonymous communication in mobile networks. The methods developed in these works
generally require a trusted administrator, as written, and their end products generally differ from ours in form and/or in statistical properties.

**ADVANTAGES OF PROPOSED SYSTEM:**

Increasing a parameter in the algorithm will reduce the number of expected rounds. However, our central algorithm requires solving a polynomial with coefficients taken from a finite field of integers modulo a prime. That task restricts the level to which can be practically raised. We show in detail how to obtain the average number of required rounds, and in the Appendix detail a method for solving the polynomial, which can be distributed among the participants.

**MODULES:**

- Homomorphic encryption Module.
- Anonymous Data Sharing
- Generalization Module.
- Cryptography Module.
MODULES DESCRIPTION:

Homomorphic encryption Module:

This module to use the first protocol is aimed at suppression-based anonymous databases, and it allows the owner of DB to properly anonymize the tuple t, without gaining any useful knowledge on its contents and without having to send to t’s owner newly generated data. To achieve such goal, the parties secure their messages by encrypting them. In order to perform the privacy-preserving verification of the database anonymity upon the insertion, the parties use a commutative and homomorphic encryption scheme.

Generalization Module:

In this module, the second protocol is aimed at generalization-based anonymous databases, and it relies on a secure set intersection protocol, such as the one found in, to support privacy-preserving updates on a generalization based k-anonymous DB.

Cryptography Module:

In this module, the process of converting ordinary information called plaintext into unintelligible gibberish called cipher text. Decryption is the reverse, in other words, moving from the unintelligible cipher text back to plaintext. A
cipher (or) cypher is a pair of algorithms that create the encryption and the reversing decryption. The detailed operation of a cipher is controlled both by the algorithm and in each instance by a key. This is a secret parameter (ideally known only to the communicants) for a specific message exchange context.

User and Admin Module:

In this module, to arrange the database based on the patient and doctor details and records. The admin to encrypt the patient reports using encryption techniques using suppression and generalization protocols.

SYSTEM CONFIGURATION:-

HARDWARE CONFIGURATION:-

- Processor - Pentium -IV
- Speed - 1.1 Ghz
- RAM - 256 MB(min)
- Hard Disk - 20 GB
- Key Board - Standard Windows Keyboard
- Mouse - Two or Three Button Mouse
- Monitor - SVGA
SOFTWARE CONFIGURATION:

- Operating System : Windows XP
- Programming Language : JAVA/J2EE
- Java Version : JDK 1.6 & above.
- Database : MYSQL