ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Anonymous (Unidentified) Communication of Data from End to End Through an Algorithm Based on Sturm's Theorem

Menavolu Rohini
M.Tech Student,
Department of CSE,
Medha Engineering College,
Khammam.

J V S Arundathi, M.Tech, Ph.D
Assistant Professor,
Department of CSE,
Medha Engineering College,
Khammam.

I. Narasimha Rao
Asst Professor & HOD,
Department of CSE,
Medha Engineering College,
Khammam.

Abstract:

Cloud computing is computing in which large groups of remote servers are networked to allow centralized data storage and online access to computer services or resources. Cloud computing relies on sharing of resources to achieve coherence and economies of scale, similar to a utility (like the electricity grid) over a network. Sharing data while preserving data and identity privacy from an untrusted cloud is still a challenging issue. For sharing private data securely among several parties an algorithm has been used. An anonymous ID assignment technique is used iteratively to assign the nodes with ID numbers ranging from 1 to N. This technique enhances data that are more complex to be shared securely. The nodes are assigned with the anonymous ID with the help of a central authority. The algorithm has been compared with the existing algorithm. New algorithm has been developed based on Sturm's theorem and Newton's identities. The number of iterations are found out with the help of Markov chain.

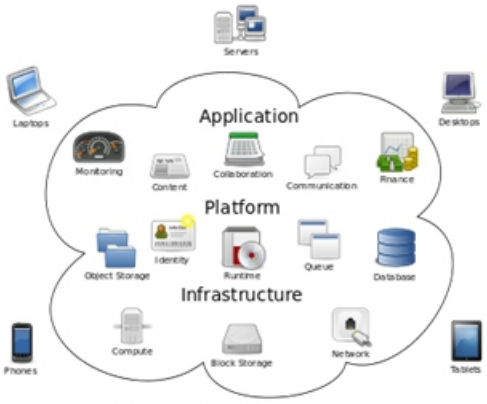
Keywords:

Privacy, encryption, cloud computing, secure sharing, identity privacy, markov chain, Anonymization, Deanonymization, sturms theorem.

Introduction:

In a cloud computing system, there's a significant workload shift. Local computers no longer have to do all the heavy lifting when it comes to running applications. The network of computers that make up the cloud handles them instead. Hardware and software demands on the user's side decrease.

The only thing the user's computer needs to be able to run is the cloud computing system's interface software, which can be as simple as a Web browser, and the cloud's network takes care of the rest.



Cloud Computing

The National Institute of Standards and Technology's definition of cloud computing identifies "five essential characteristics":On-demand self-service: A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.Broad network access: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).

Resource pooling:

The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Rapid elasticity:

Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear unlimited and can be appropriated in any quantity at any time.

Measured service:

Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

Sturm's theorem:

In mathematics, the Sturm's sequence of a univariate polynomial p is a sequence of polynomials associated with p and its derivative by a variant of Euclid's algorithm for polynomials. Sturm's theorem expresses the number of distinct real roots of p located in an interval in terms of the number of changes of signs of the values of the Sturm's sequence at the bounds of the interval. Applied to the interval of all the real numbers, it gives the total number of real roots of p.

Whereas the fundamental theorem of algebra readily yields the overall number of complex roots, counted with multiplicity, it does not provide a procedure for calculating them. Sturm's theorem counts the number of distinct real roots and locates them in intervals. By subdividing the intervals containing some roots, it can isolate the roots into arbitrary small intervals, each containing exactly one root. This yields an arbitrary-precision numeric root finding algorithm for univariate polynomials.

Sturm Chain:

A Sturm chain or Sturm sequence is a finite sequence of polynomials

$$p_0, p_1, \ldots, p_m$$

of decreasing degree with these following properties:

- po = p is square free (no square factors, i.e., no repeated roots);
- if p(ξ) = o, then sign(p1(ξ)) = sign(p□(ξ));
- if pi(ξ) = o for o < i < m then sign(pi−1(ξ)) = −sign(pi+1(ξ));
- pm does not change its sign.

Sturm's sequence is a modification of Fourier's sequence.To obtain a Sturm chain, Sturm himself proposed to choose the intermediary results when applying Euclid's algorithm to p and its derivative:

$$\begin{aligned}
p_0(x) &:= p(x), \\
p_1(x) &:= p'(x), \\
p_2(x) &:= -\operatorname{rem}(p_0, p_1) = p_1(x)q_0(x) - p_0(x), \\
p_3(x) &:= -\operatorname{rem}(p_1, p_2) = p_2(x)q_1(x) - p_1(x), \\
&\vdots \\
0 &= -\operatorname{rem}(p_{m-1}, p_m),
\end{aligned}$$

where rem(pi, pj) and qi are the remainder and the quotient of the polynomial long division of pi by pj, and where m is the minimal number of polynomial divisions (never greater than deg(p)) needed to obtain a zero remainder. That is, successively take the remainders with polynomial division and change their signs. Since deg(pi+1) < deg(pi) foro ≤ i < m, the algorithm terminates. The final polynomial, pm, is the greatest common divisor of p and its derivative. If p is square free, it shares no roots with its derivative, hencepm will be a non-zero constant polynomial. The Sturm chain, called the canonical Sturm chain, then is

$$p_0, p_1, p_2, \ldots, p_m.$$

If *p* is not square-free, this sequence does not formally satisfy the definition of a Sturm chain above; nevertheless it still satisfies the conclusion of Sturm's theorem (below).

Statement:

Let $p = p_o \ldots p_m$ be a Sturm chain, where *p* is a square-free polynomial, and let $\sigma(\xi)$ denote the number of sign changes (ignoring zeroes) in the sequence

$$p_0(\xi), p_1(\xi), p_2(\xi), \ldots, p_m(\xi).$$

Sturm's theorem then states that for two real numbers $a < b$, the number of distinct roots of *p* in the half-open interval $(a, b]$ is $\sigma(a) - \sigma(b)$.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

EXISTING SYSTEM:

Existing and new algorithms for assigning anonymous IDs are examined with respect to trade-offs between communication and computational requirements.. Also, suppose that access to the database is strictly controlled, because data are used for certain experiments that need to be maintained confidential. Clearly, allowing Alice to directly read the contents of the tuple breaks the privacy of Bob; on the other hand, the confidentiality of the database managed by Alice is violated once Bob has access to the contents of the database. Thus, the problem is to check whether the database inserted with the tuple is still k-anonymous, without letting Alice and Bob know the contents of the tuple and the database respectively.

Disadvantages:

1.The database with the tuple data does not be maintained confidentially.

2.The existing systems another person to easily access database.

PROPOSED SYSTEM:

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 trustedcentral authority.

Advantages:

1.The anonymity of DB is not affected by inserting the records.

2.We provide security proofs and experimental results for both protocols.

MODULES:

1.Homomorphic encryption Module.

2.Generalization Module.

3.Cryptography Module.

4.User and Admin Module.

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.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

The admin to encrypt the patient reports using encryption techniques using suppression and generalization protocols.

Algorithms Used:

1) Secure Sum

2) Anonymous Data Sharing With Power Sums

3) Find Aida

Algorithm 1 (Secure Sum): Given nodes $n_1, \ldots, n_N$ each holding an data item d_i from a finitely representable abelian group, share the value $T = \sum d_i$ among the nodes without revealing the values d_i.

1) Each node n_i, $i = 1, \ldots, N$ chooses random values $r_{i,1}, \ldots, r_{i,N}$ such that

$$r_{i,1} + \cdots + r_{i,N} = d_i$$

2) Each "random" value $r_{i,j}$ is transmitted from node n_i to node n_j. The sum of all these random numbers $r_{i,j}$ is, of course, the desired total T.

3) Each node n_j totals all the random values received as:

$$s_j = r_{1,j} + \cdots + r_{N,j}$$

4) Now each node n_i simply broadcasts s_i to all other nodes so that each node can compute:

$$T = s_1 + \cdots + s_N$$

Algorithm 2 (Anonymous Data Sharing With Power Sums): Given nodes $n_1, \ldots, n_N$ each holding a data item d_i from a finitely representable field F, make their data items public to all nodes without revealing their sources.

1) Each node n_i computes d_i^n over the field F for $n = 1, 2, \ldots, N$. The nodes then use secure sum to share knowledge of the power sums:

$P_1 = \sum_{i=1}^{N} d_i^1$	$P_2 = \sum_{i=1}^{N} d_i^2$	$\cdots$	$P_N = \sum_{i=1}^{N} d_i^N$

2) The power sums $P_1, \ldots, P_N$ are used to generate a polynomial which has $d_1, \ldots, d_N$ as its roots using Newton's Identities as developed in [30]. Representing the Newton polynomial as

$$p(x) = c_N x^N + \cdots + c_1 x + c_0 \quad (1)$$

the values $c_0, \ldots, c_N$ are obtained from the equations:

$$\begin{aligned}
c_N &= -1 \\
c_{N-1} &= -\frac{1}{1}(c_N P_1) \\
c_{N-2} &= -\frac{1}{2}(c_{N-1} P_1 + c_N P_2) \\
c_{N-3} &= -\frac{1}{3}(c_{N-2} P_1 + c_{N-1} P_2 + c_N P_3) \\
c_{N-4} &= -\frac{1}{4}(c_{N-3} P_1 + c_{N-2} P_2 + c_{N-1} P_3 + c_N P_4) \cdots \\
c_{N-m} &= -\frac{1}{m} \sum_{k=1}^{m} c_{N-m+k} P_k \quad (2)
\end{aligned}$$

3) The polynomial $p(x)$ is solved by each node, or by a computation distributed among the nodes, to determine the roots $d_1, \ldots, d_N$.

The power sums P_i can be collected and shared using a single round of secure sum by sending them as an array and applying the method to the vectors transmitted and received. The power sums are symmetric functions, and thus no association is made between n_i and the value of d_i. However, nonetheless, the information contained in these sums can be used to find the values of the data items $d_1, \ldots, d_N$.

The choice $c_N = -1$, chosen for consistency with [31], may be replaced by $c_N = 1$ or any other nonzero value. Also, note that in the typical $F = GF(P)$ case, the solution for the c_i requires finding the multiplicative inverse of the coefficients $2, 3, 4, \ldots, N$ modulo P. While the Euclidean algorithm could be used, the inverses $1/x$ can easily be computed in the order $x = 1, 2, \ldots, N$ by the formulae:

$$q = P/x + r; \quad 1/x = -q\,(1/r) \pmod{P}$$

After the integer division with remainder r, $1/r$ will already be known, since $r < x$.

Algorithm 3 (Find AIDA): Given nodes $n_1, \ldots, n_N$, use distributed computation (without central authority) to find an anonymous indexing permutation $s : \{1, \ldots, N\} \rightarrow \{1, \ldots, N\}$.

1) Set the number of assigned nodes $A = 0$.
2) Each unassigned node n_i chooses a random number r_i in the range 1 to S. A node assigned in a previous round chooses $r_i = 0$.
3) The random numbers are shared anonymously. One method for doing this was given in Section III. Denote the shared values by $q_1, \ldots, q_N$.
4) Let $q_1, \ldots, q_k$ denote a revised list of shared values with duplicated and zero values entirely removed where k is the number of unique random values. The nodes n_i which drew unique random numbers then determine their index s_i from the position of their random number in the revised list as it would appear after being sorted:

$$s_i = A + \text{Card}\{q_j : q_j <= r_i\}$$

5) Update the number of nodes assigned: $A = A + k$.
6) If $A < N$ then return to step (2).

(Source: Larry A. Dunning, Member, IEEE, and Ray Kresman-"Privacy Preserving Data Sharing With Anonymous ID Assignment"-IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 8, NO. 2, FEBRUARY 2013.)

CONCLUSION:

There are several variants of AIDA algorithm can be used but here we make use of AIDA algorithm along with the central authority therefore merged with the concept of encoding and decoding for improved security purposes. Other algorithms can also be used and compared for detailed study of the behaviour of the system. Here encoding and decoding methodologies largely decrease the number of rounds required for ID assignment and there by decreasing the overheads and enhancing the performance of the entire system. As all the nodes are assumed to be dishonest, using this algorithm helps to continue with the reliable data sharing.

REFERENCES:

[1] Larry A. Dunning, Member, IEEE, and Ray Kresman, "Privacy preserving data sharing with anonymous ID assignment", IEEE transactions on information forensics and security, vol. 8, no. 2, february 2013

[2] Sarbanes–Oxley Act of 2002, Title 29, Code of Federal Regulations, Part 1980, 2003.

3] J. Wang, T. Fukasama, S. Urabe, and T. Takata, "A collusion-resistant approach to privacypreserving distributed data mining," IEICE Trans. Inf. Syst. (Inst. Electron. Inf. Commun. Eng.), vol.E89-D, no. 11, pp. 2739–2747, 2006.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

[4] A. Shamir, "How to share a secret," Commun. ACM, vol. 22, no. 11, pp. 612–613, 1979.

[5] A. Friedman, R. Wolff, and A. Schuster, "Providing k-anonymity in data mining," VLDB Journal,vol. 17, no. 4, pp. 789–804, Jul. 2008.

[6] F. Baiardi, A. Falleni, R. Granchi, F. Martinelli, M. Petrocchi, and A. Vaccarelli, "Seas, a secure e-voting protocol: Design and implementation," Comput. Security, vol. 24, no. 8, pp. 642–652, Nov.2005.

[7] D. Chaum, "Untraceable electronic mail, return address and digital pseudonyms," Commun.ACM, vol. 24, no. 2, pp. 84–88, Feb. 1981.

[8] Q. Xie and U. Hengartner, "Privacy-preserving matchmaking for mobile social networking secure against malicious users," in Proc. 9th Ann. IEEE Conf. Privacy, Security and Trust, Jul. 2011, pp.252–259.

[9] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game," in Proc. 19th Ann.ACM Conf. Theory of Computing, Jan. 1987, pp. 218–229, ACM Press.

[10] A. Yao, "Protocols for secure computations," in Proc. 23rd Ann. IEEE Symp. Foundations of Computer Science, 1982, pp. 160–164, IEEE Computer Society