Enhancing data mining techniques for secured data sharing and privacy preserving on web mining

Miss Snehal K. Dekate*, Prof. Jayant Adhikari, Prof. Sulbha Parate*****

* Computer Science & Engg.
Tulsiramji Gaykwad Patil College of Engineering & Technology
Rashtrasant Tukadoji Maharaj Nagpur University

Abstract- *The enhancing data techniques are used in the user database for secure their database from other unauthorized user. This technique is useful for privacy preserving; securely share data among N number of parties. And also apply data mining approach on web service.*

Keyword*- Data Mining,Privacy Preserving,Security, Web Services*

I. INTRODUCTION

Algorithms for assigning anonymous IDs are examined with respect to threshold 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.

The popularity of internet as a communication medium whether for personal or business use depends in part on its support for anonymous communication. Businesses also have legitimate reasons to engage in anonymous communication and avoid the consequences of identity revelation. For example, to allow dissemination of summary data without revealing the identity of the entity the underlying data is associated with, or to protect whistle-blower's right to be anonymous and free from political or economic retributions.

Each algorithm can be reasonably implemented and each has its advantages. Our use of the Newton identities greatly decreases communication overhead. This can enable the use of a larger number of "slots" with a consequent reduction in the number of rounds required. The solution of a polynomial can be avoided at some expense by using Sturm's theorem. The development of a result similar to the Sturm's method over a finite field is an enticing possibility.

II. REVIEW OF LITERATURE

In proposed system algorithms are secure in an information theoretic sense. Provide security for sharing data using AES and RSA algorithm. Existing and new algorithms for assigning anonymous IDs areexamined with respect to trade-offs between communication andcomputational requirements. The new algorithms are built on topof a secure sum data mining operation. (K-means).

AES Algorithm

The AES cipher like DES, AES is a symmetric block cipher. This means that it uses the same key for both encryption and decryption. However, AES is quite different from DES in a number of ways. The algorithm Rijndael allows for a variety of block and key sizes and not just the 64 and 56 bits of DES' block and key size. The block and key can in fact be chosen independently from 128,160,192,224,256 bits and need not be the same. However, the AES standard states that the algorithm can only accept a block size of 128 bits and a choice of three keys -128,192,256 bits. Depending on which version is used, the name of the standard is modified to AES-128, AES-192 or AES-256 respectively. As well as these differences AES differs from DES in that it is not a feistel structure. Recall that in a feistel structure, half of the data block is used to modify the other half of the data block and then the halves are swapped. In this case the entire data block is processed in parallel during each round using substitutions and permutations.

A number of AES parameters depend on the key length. For example, if the key size used is 128 then the number of rounds is 10 whereas it is 12 and 14 for 192 and 256 bits respectively. At present the most common key size likely to be used is the128 bit key.This description of the AES algorithm therefore describes this particular

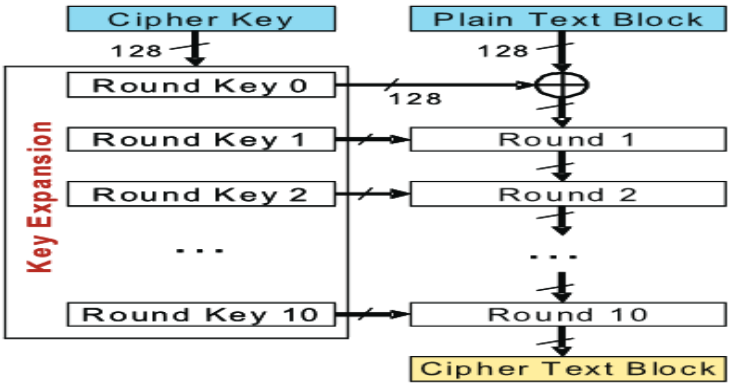


Figure 1. AES Algorithm Structure

RSA Algorithm

RSA is one of the first practicable public-key cryptosystems and is widely used for secure data transmission. In such a cryptosystem, the encryption key is public and differs from the decryption key which is kept secret. In RSA, this asymmetry is based on the practical difficulty of factoring the product of two large prime numbers, the factoring problem. RSA stands for Ron Rivest, Adi Shamir and Leonard Adleman, who first publicly described the algorithm in 1977. Clifford Cocks, an English mathematician, had developed an equivalent system in 1973, but it wasn't declassified until 1997.[1]

A user of RSA creates and then publishes a public key based on the two large prime numbers, along with an auxiliary value. The prime numbers must be kept secret. Anyone can use the public key to encrypt a message, but with currently published methods, if the public key is large enough, only someone with knowledge of the prime numbers can feasibly decode the message.[2] Breaking RSA encryption is known as the RSA problem. It is an open question whether it is as hard as the factoring problem

Key generation

RSA involves a *public key* and a *private key*. The public key can be known by everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted in a reasonable amount of time using the private key. The keys for the RSA algorithm are generated the following way:

1. Choose two distinct prime numbers$p$ and q.
   - For security purposes, the integers p and q should be chosen at random, and should be of similar bit-length. Prime integers can be efficiently found using a primality test.
2. Compute $n = pq$.
   - n is used as the modulus for both the public and private keys. Its length, usually expressed in bits, is the key length.
3. Compute $\varphi(n) = \varphi(p)\varphi(q) = (p - 1)(q - 1) = n - (p + q - 1)$, where φ is Euler's totient function.
4. Choose an integer e such that $1 < e < \varphi(n)$ and $\gcd(e, \varphi(n)) = 1$; i.e., e and $\varphi(n)$ are coprime.
   - e is released as the public key exponent.
   - e having a short bit-length and small Hamming weight results in more efficient encryption – most commonly $2^{16} + 1 = 65,537$. However, much smaller values of e (such as 3) have been shown to be less secure in some settings.[5]
5. Determine d as $d \equiv e^{-1} \pmod{\varphi(n)}$; i.e., d is the multiplicative inverse of e (modulo $\varphi(n)$).

   - This is more clearly stated as: solve for d given $d \cdot e \equiv 1 \pmod{\varphi(n)}$
   - This is often computed using the extended Euclidean algorithm. Using the pseudocode in the *Modular integers* section, inputs a and n correspond to e and $\varphi(n)$, respectively.
   - d is kept as the private key exponent.

The *public key* consists of the modulus n and the public (or encryption) exponent e. The *private key* consists of the modulus n and the private (or decryption) exponent d, which must be kept secret. p, q, and $\varphi(n)$ must also be kept secret because they can be used to calculate d.

- An alternative, used by PKCS#1, is to choose d matching $de \equiv 1 \pmod{\lambda}$ with $\lambda = \text{lcm}(p-1, q-1)$, where lcm is the least common multiple. Using λ instead of $\varphi(n)$ allows more choices for d. λ can also be defined using the Carmichael function, $\lambda(n)$.
- The ANSI X9.31 standard prescribes, IEEE 1363 describes, and PKCS#1 allows, that p and q match additional requirements: being strong primes, and being different enough that Fermat factorization fails.

Encryption

Alice transmits her public key (n, e) to Bob and keeps the private key d secret. Bob then wishes to send message M to Alice.

He first turns M into an integer m, such that $0 \leq m < n$ by using an agreed-upon reversible protocol known as a padding scheme. He then computes the ciphertext c corresponding to

$$c \equiv m^e \pmod{n}$$

This can be done efficiently, even for 500-bit numbers, using Modular exponentiation. Bob then transmits c to Alice.

Note that at least nine values of m will yield a ciphertext c equal to m,[note 1] but this is very unlikely to occur in practice.

Decryption

Alice can recover m from c by using her private key exponent d via computing

$$m \equiv c^d \pmod{n}$$

Given m, she can recover the original message M by reversing the padding scheme.

(In practice, there are more efficient methods of calculating c^d using the precomputed values below.)

K-Means

K-Means clustering generates a specific number of disjoint, flat (non-hierarchical) clusters. It is well suited to generating globular clusters. The K-Means method is numerical, unsupervised, non-deterministic and iterative.

K-Means Algorithm Properties

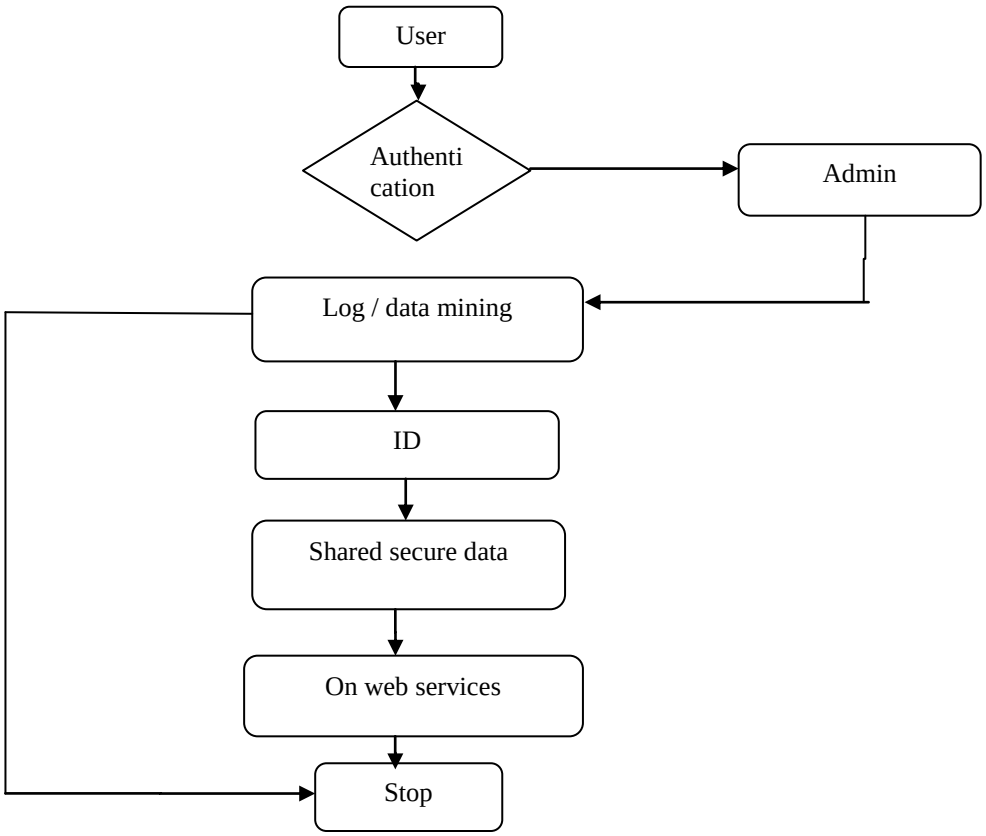
- There are always K clusters.
- There is always at least one item in each cluster.
- The clusters are non-hierarchical and they do not overlap.
- Every member of a cluster is closer to its cluster than any other cluster because closeness does not always involve the 'center' of clusters.

The K-Means Algorithm Process

- The dataset is partitioned into K clusters and the data points are randomly assigned to the clusters resulting in clusters that have roughly the same number of data points.
- For each data point:
- Calculate the distance from the data point to each cluster.

- If the data point is closest to its own cluster, leave it where it is. If the data point is not closest to its own cluster, move it into the closest cluster.
- Repeat the above step until a complete pass through all the data points results in no data point moving from one cluster to another. At this point the clusters are stable and the clustering process ends.
- The choice of initial partition can greatly affect the final clusters that result, in terms of inter-cluster and intracluster distances and cohesion.

III. PROJECT PLANNING



IV. CONCLUSION

In this report, we presented a survey of the broad areas of privacy-preserving data mining and the underlying algorithms. Data modification techniques such as k -Means based techniques. We discussed methods for privacy-preserving mining.Further, we discussed some fundamental limitations of the problem of privacy-preservation in presence of increased amounts of public information and background knowledge. We presented a set of experimental results and also analyzed them from the perspective of data privacy and data utilization. We have also presented a number of diverse application domains for which privacy-preserving data mining methods are useful. Finally, we identified few areas which require further research efforts in the domain of privacy-preserving data mining.

V. 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] Anita Rajendra Zope, Amarsinh Vidhate, and Naresh Harale "Data Mining Approach in Security Information and Event Management" International Journal of Future Computer and Communication, Vol. 2, No. 2, April 2013

[3] Pasupuleti Rajesh and Gugulothu Narsimha "PRIVACY PRESERVING DATA MINING BY USING IMPLICIT FUNCTION THEOREM", International Journal of Network Security & Its Applications (IJNSA), Vol.5, No.2, March 2013

[4] Jianjun Duan, Joe Hurd, Guodong Li,Scott Owens, Konrad Slind, and Junxing Zhang "Functional Correctness Proofs of Encryption Algorithms"

[5] S. Subramaniam, T. Palpanas, D. Papadopoulos,V. Kalogeraki, and D. Gunopulos, "Online outlier detection in sensor data using non-parametric models, in VLDB", 2006

[6] Lu-An Tang_, Jiawei Han, and Guofei Jiang, "Mining Sensor Data in Cyber-Physical Systems" TSINGHUA SCIENCE AND TECHNOLOGY ISSN 1007-0214 01/11 pp225-234 Volume 19, Number 3, June 2014

[7] Qi Xie and Urs Hengartner, "Privacy-Preserving Matchmaking For Mobile Social Networking Secure Against Malicious Users", 2011 Ninth Annual International Conference on Privacy, Security and Trust

[8] Chris Clifton, Murat Kantarcioglu, Jaideep Vaidya, Xiaodong Lin, Michael Y. Zhu, "Tools for Privacy Preserving Distributed Data Mining"

[9] Dr.R. Sugumar1, Dr.A. Rengarajan 2, M.Vijayanand3, "Extending K-Anonymity to Privacy Preserving Data Mining Using Association Rule Hiding Algorithm"

AUTHORS

First Author – Miss Snehal K. Dekate, M.Tech (CSE), dekatesnehal90@gmail.com.
Second Author – Prof. Jayant Adhikari, adhikari.jayant@gmail.com.
Third Author – Prof Sulbha Prate, sulbha.cse@tgpcet.com.