International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 7 | ISSN : 2456-3307

Preservation of Online Users' Keyword Search Data Using Asymmetric Algorithm Method

R. Anusha[1*] , D. Sudha[2]
[1*]Research Scholar, Department of Computer Science, A.V.C College, Mayiladuthurai, India
[2]Associate Professor, Department of Computer Science, A.V.C College, Mayiladuthurai, India

ABSTRACT

Internet is the place where tons of online user behavior data are generated every day. These data are used to extract users' valuable information for research purposes or business interests. But, the data are under the risk of being exposed to third parties. Methods are implemented to perform the data aggregation in a privacy preserving manner. Most of the available methods assure strong privacy protection at the cost of very limited aggregation such as only summation, which hardly satisfies the need of behavior analysis. In this paper, proposed a model called PPSA. This model encrypts the users' sensitive data to prevent privacy from both outside analysts and the aggregation service provider. Also, completely supports selective aggregate functions for online user behavior analysis and guaranteeing differential privacy. Homomorphic RSA algorithm is used for encrypting users' online behavior data. Implementation is done and its performances are evaluated based on a real time behavior set. Experimental results show that the proposed method effectively supports both overall aggregate queries and various selective aggregate queries with acceptable computation and communication overheads.

Keywords : RSA, Homomorphic, Privacy Preserving, Selective Aggregation

I. INTRODUCTION

The Privacy-preserving data aggregation problem has long been a hot research issue in the field of applied cryptography. In numerous real life applications such as crowd sourcing or mobile cloud computing, individuals need to provide their sensitive data (location-related or personal-information-related) to receive specific services from the entire system (e.g., location based services or mobile based social networking services). There are usually two different models in this problem: 1) an external aggregator collects the data and wants to conduct an aggregation function on participants' data (e.g., crowd sourcing); 2) participants themselves are willing to jointly compute a specific aggregation function whose input data is co-provided by them (e.g., social networking services). However, the individual's data should be kept secret, and the aggregator or other participants are not supposed to learn any useful information about it. Secure Multi-party Computation (SMC), Homomorphic Encryption (HE) and other cryptographic methodologies can be partially or fully exploited to solve this problem, but they are subject to some restrictions in this problem. Secure Multi-party Computation (SMC) was first formally introduced in 1982 as Secure Two-Party Computation. Homomorphic Encryption (HE) allows direct addition and multiplication of ciphertexts while preserving decryptability. That is, Enc (m1) $\otimes$ Enc (m2) = Enc (m1 $\times$ m2), where Enc (m) stands for the ciphertext of m, and $\otimes$, $\times$ refer to the homomorphic operations on the ciphertext and plaintexts respectively. One could also try to solve this problem using this technique, but HE uses the same decryption key for original data and the aggregated data. That is, the operator who executes homomorphic operations upon the ciphertexts are not authorized to achieve the final result. This forbids aggregator from decrypting the aggregated result, because if the aggregator is allowed to decrypt the final result, he can also decrypt the individual ciphertext received, which contradicts this motivation. Also, because the size of the plaintext space is limited, the number of addition and multiplication operations executed upon ciphertexts was limited until a fully homomorphic encryption scheme is proposed and implemented it in.

However, the complexity of general HE is too high to use in real application. A HE scheme is proposed which sacrificed possible number of multiplications for speed, but it still needs too much time to execute homomorphic operations on ciphertexts. Besides the aforementioned drawbacks, both SMC and HE require an initialization phase during which participants request keys from key issuers via secure channel. This could be a security hole since the security of those schemes relies on the assumption that keys are disclosed to authorize participants only. The explosive growth of hardware, software along with immense computing and communication power of the system and devices, made it unbelievably easy to store, retrieve and process large amounts of information. Good amount of privacy issues also arise with the proliferation of digital technologies. Explosive progress in networking, storage, and processor technologies has led to the creation of ultra large databases that record unprecedented amount of transactional information. In tandem with this dramatic increase in digital data, concerns about informational privacy have emerged globally. Privacy issues are further exacerbated now that the World Wide Web (WWW) makes it easy for the new data to be automatically collected and added to databases. With ubiquitous connectivity, people are increasingly using electronic technologies in business-to-consumer and business-to-business settings. This in effect helps a third party to acquire the confidential and private information from various avenues. Depending upon the nature of the information, users may not be willing to divulge the individual values of records. This has lead to concerns that the private data may be misused for a variety of purposes. Privacy can be defined as the limited access to a person or a process and to all the features related to the person or the process. Privacy preservation is important from both individual as well as organizational perspectives. For example, customers might send to a remote database queries that contain private information. Two competing commercial organizations might jointly invest in a project that must satisfy both organizations' private and valuable constraints, and so on. In order to alleviate these concerns, a number of techniques have recently been proposed to perform the data mining tasks in a privacy-preserving way, which is called Privacy Preserving Data Mining (PPDM). The research of PPDM is aimed at bridging the gap between collaborative data mining and data privacy. Privacy-preserving data mining finds numerous applications in surveillance, in-network processing, which are naturally supposed to be “privacy-violating” applications.

The key is to design methods which continue to be effective, without compromising security. In this paper, consider a scenario where data aggregation needs to be done in privacy-preserved way for distributed computing platform. There are number of data sources which collect or produce data. The data collected or produced by the sources is private and the owner or the source does not like to reveal the content of the data. But the collected data from the source is to be aggregated by an aggregator, which may be a third party or part of the network, where the data sources belong. strategic decisions, rather than minimizing the distortion of all statistics.

In other words, the goal here is not only to protect personally identifiable information but also some patterns and trends that are not supposed to be discovered.

Privacy Preservation in Data Mining has some limitations: Privacy Preservation Data Mining techniques do not mean pefect privacy, for example, The SMC computation won't reveal the sensitive data, but the data mining result will enable all parties to estimate the value of the sensitive data. It isn't that the SMC was "broken", but that the result itself violates privacy.

Models of PPDM

In the study of privacy-preserving data mining (PPDM), there are mainly four models as
follows:

1. Trust Third Party Model

The goal standard for security is the assumption that we have a trusted third party to whom we can give all data. The third party performs the computation and delivers only the results – except for the third party, it is clear that nobody learns anything not inferable from its own input and the results. The goal of secure protocols is to reach this same level of privacy preservation, without the problem of finding a third party that everyone trusts.

2. Semi-honest Model

In the semi-honest model, every party follows the rules of the protocol using its correct input, but after the protocol is free to use whatever it sees during execution of the protocol to compromise security.

3. Malicious Model

In the malicious model, no restrictions are placed on any of the participants. Thus any party is completely free to indulge in whatever actions it pleases. In general, it is quite difficult to develop efficient protocols that are still valid under the malicious model. However, the semi-honest model does not provide sufficient protection for many applications.

4. Other Models - Incentive Compatibility

While the semi-honest and malicious models have been well researched in the cryptographic community, other models outside the purview of cryptography are possible. One example is the interesting economic notion of incentive compatibility. A protocol is incentive compatible if it can be shown that a cheating party is either caught or else suffers an economic loss. Under the rational model of economics, this would serve to ensure that parties do not have any advantage by cheating. Of course, in an irrational model, this would not work. We remark, in the "real world", there is no external party that can be trusted by all parties, so the Trust Third Party Model is a ideal model.

Evaluation of privacy preserving algorithms

An important aspect in the development and assessment of algorithms and tools, for privacy preserving data mining is the identification of suitable evaluation criteria and the development of related benchmarks. It is often the case that no privacy preserving algorithm exists that outperforms all the others on all possible criteria. Rather, an algorithm may perform better that another one on specific criteria, such as performance and/or data utility. It is thus important to provide users with a set of metrics which will enable them to select the most appropriate privacy preserving technique for the data at hand, with respect to some specific parameters they are interested in optimizing.

The rest of the journal is summarized as follows: Chapter II deals with Related work. Chapter III explains the proposed method. Chapter IV discusses about the results of the paper. Chapter V is the conclusion.

II. RELATED WORK

In [1], Vibhor et al., proposes PASTE, the first differentially private aggregation algorithm for distributed time-series data that offers good practical utility without any trusted server. PASTE addresses two important challenges in participatory data-mining applications where (i) individual users collect temporally correlated time-series data (such as location traces, web history, personal health data), and (ii) an untrusted third-party aggregator wishes to run aggregate queries on the data. To address this, PASTE incorporates two new algorithms. To ensure differential privacy for time-series data despite the presence of temporal correlation, PASTE uses the Fourier Perturbation Algorithm (FPAk).

In [2], Taeho et al., considers consider how an external aggregator or multiple parties can learn some algebraic statistics. Also proposes several protocols that successfully guarantee data privacy under this weak assumption while limiting both the communication and computation complexity of each participant to a small constant.

In [3], Peter et al., provides answer for the fundamental question of characterizing the level of overall privacy degradation as a function of the number of queries and the privacy levels maintained by each privatization mechanism. Our solution is complete: we prove an upper bound on the overall privacy level and construct a sequence of privatization mechanisms that achieves this bound.

In [4], Armstrong et al., discusses the design and implementation of geographical masks that not only preserve the security of individual health records, but also support the investigation of questions that can be answered only with some knowledge about the location of health events. We describe several alternative methods of masking individual-level data, evaluate their performance, and discuss both the degree to which we can analyze masked data validly as well as the relative security of each approach, should anyone attempt to recover the identity of an individual from the masked data.

In [5], Arijit et al., developed a scheme to provide privacy preservation in a much simpler way with the help of a secure key management scheme and randomized data perturbation technique. We consider a scenario in which two or more parties owning confidential data need to share only for aggregation purpose to a third party, without revealing the content of the data. Through simulation results the efficacy of our scheme is shown by comparing the results with one of the established schemes.

In [6], Nathan et al., present a method to convert learned neural networks to CryptoNets, neural networks that can be applied to encrypted data. This allows a data owner to send their data in an encrypted form to a cloud service that hosts the network. The encryption ensures that the data remains confidential since the cloud does not have access to the keys needed to decrypt it. Nevertheless, we will show that the cloud service is capable of applying the neural network to the encrypted data to make encrypted predictions, and also return them in encrypted form. These encrypted predictions can be sent back to the owner of the secret key who can decrypt them. Therefore, the cloud service does not gain any information about the raw data nor about the prediction it made.

In [7], Ihsan et al., explains the concept of cloud computing receiving a great deal of attention both in publication and among users. Cloud computing is the delivery of computing services over the Internet. Cloud services allow individuals and businesses to use software and hardware resources that are managed by cloud providers at remote locations. The distance between the client and the physical location of his data creates a barrier because this data can be accessed by a third party and this would affect the privacy of client's data. The using of traditional encryption schemes to encrypt the remoted data before sending to the cloud provider has been most widely used technique to bridge this security gab. But, the client will need to provide the private key to the server to decrypt the data before perform the calculations required. Homomorphic encryption allows performing computations on encrypted data without decryption. This paper deals with the use of homomorphic encryption to encrypt the client's data in cloud server and also it enables to execute required computations on this encrypted data.

In [8], Jung et al., suggests a method to construct a homomorphic encryption scheme for approximate arithmetic. It supports approximate addition and multiplication of encrypted messages, together with the rescaling procedure for managing the magnitude of plaintext. This procedure truncates a ciphertext into a smaller modulus, which results in rounding of plaintext after homomorphic operations. The main idea is to place a noise after the significant figures of message. This noise is added to the plaintext for security, but considered to be part of error occurred during approximate computations, which is reduced along with plaintext by rescaling. As a result, our decryption structure outputs an approximate value of plaintext with the predetermined precision. Also proposed a new batching method for RLWE-based construction. A plaintext polynomial of characteristic zero is mapped to a message vector of complex numbers via complex canonical embedding map which is an isometric ring homomorphism. The size of error is preserved during transformation and so it enables us to remove the errors after decryption procedure. Our construction has the bit size of ciphertext modulus linear in the circuit depth due to rescaling procedure while all the previous works either require exponentially large size of modulus or expensive computations such as bootstrapping or bit extraction. One important feature of our method is that the precision loss during evaluation is bounded by circuit depth and it is at most one more bit compared with unencrypted approximate arithmetic such as floating-point operations. In addition to the basic approximate circuits, we show that our scheme can be applied to efficiently evaluate the transcendental functions such as multiplicative inverse, exponential function, logistic function, and discrete Fourier transform.

In [9], Sungwook et al., propose the first privacy-preserving matrix factorization using fully homomorphic encryption. On inputs of encrypted users' ratings, our protocol performs matrix factorization over the encrypted data and returns encrypted outputs so that the recommendation system knows nothing on rating values and resulting user/item profiles. It provides a way to obfuscate the number and list of items a user rated without harming the accuracy of recommendation, and additionally protects recommender's tuning parameters for business benefit and allows the recommender to optimize the parameters for quality of service. To overcome performance degradation caused by the use of fully homomorphic encryption, we introduce a novel data structure to perform computations over encrypted vectors, which are essential operations for matrix factorization, through secure 2-party computation in part. With the data structure, the proposed protocol requires dozens of times less computation cost over those of previous works.

In [10] Ana et al., compares the NTRU and BGV schemes in their non-scale invariant (messages in the lower bits), and their scale invariant (message in the upper bits) forms. The scale invariant versions are often called the YASHE and FV schemes. As an additional optimization, we also investigate the effect of modulus reduction on the scale-invariant schemes. We compare the schemes using the "average case" noise analysis presented. In addition we unify notation and techniques so as to show commonalities between the schemes. We find that the BGV scheme appears to be more efficient for large plaintext module, whilst YASHE seems more efficient for small plaintext module.

III. PROPOSED METHOD

The past two decades have seen large increases in the size and number of computer-based systems data. Much of this information, however, is confidential. It is furnished by people with the assurance that it not be made available in a form from which the identity of any one of them can be determined. Access to such information normally takes place under one of three modes. The first is when a data user is a member of the organization that is the custodian of the data. Here the user may access the data in all of its detail but must ensure that when results of analyses are disclosed, individual-level information is not released, nor is any other information from which one can infer individual-level data. The second mode is when a data user who does not belong to the custodial organization applies for and is granted privileged access to the information. The third mode of access is when information is released to the general public in a form that protects the security of all individual records. In this work, proposed a model called PPSA. This model encrypts the users' sensitive data to prevent privacy from both outside analysts and the aggregation service provider. Also, completely supports selective aggregate functions for online user behavior analysis and guaranteeing differential privacy. At their best, differentially private database mechanisms can make confidential data widely available for accurate data analysis, without resorting to data clean rooms, data usage agreements, data protection plans, or restricted views. Homomorphic RSA algorithm is used for encrypting users' online behavior data. Implementation is done and its performances are evaluated based on a real time behavior set. Experimental results show that the proposed method effectively supports both overall aggregate queries and various selective aggregate queries with acceptable computation and communication overheads.

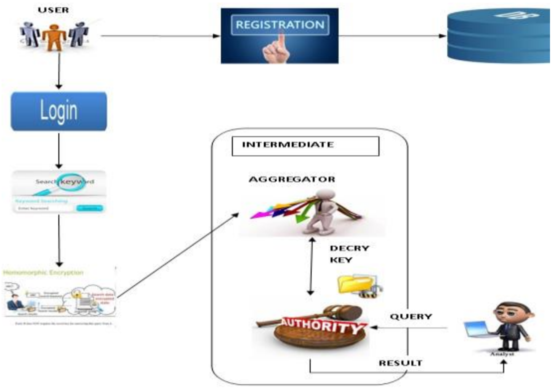


Figure 1: Architecture Diagram

FULLY HOMOMORPHIC ENCRYPTION

A fully homomorphic encryption system enables computations to be performed on encrypted data without needing to first decrypt the data. Such cryptosystems have natural applications in secure, privacy-preserving computation as well as many other areas.

A FHE scheme consists of five algorithms as follows:

1. Key generation (KG): The algorithm takes as an input a security parameter k and outputs a public and private key pair (pk, sk), where pk is public, while sk is kept secret.

2. Encryption (E): The algorithm takes as input a plaintext m ε {0,1} and the public key pk and output a ciphertext c , denoted as c ≡ E(m,pk).

3. Decryption (D): The algorithm takes as input a ciphertext c and the private key sk, and outputs a plaintext m ε {0,1}, denoted as m ≡ D(c,pk).

4. Homomorphic addition (Add): The algorithm takes as input two ciphertexts $c_1 \equiv E(m_1,pk)$, $c_2 \equiv E(m_2,pk)$ and the public key pk and outputs a ciphertext c, denoted as $c \equiv Add(c_1, c_2, pk) \equiv c_1 + c_2$, such that D(c, sk) = m1+ m2.

5. Homomorphic multiplication (Mult): The algorithm takes as input two ciphertexts $c_1 \equiv E(m_1,pk)$, $c_2 \equiv E(m_2,pk)$ and the public key and outputs a ciphertext $c = Mult(c_1, c_2, pk) = c_1 * c_2$, such that D(c, sk) = m1* m2.

A FHE scheme (KG,E,D,Add,Mult) is semantically secure if, given any public key pk, no probabilistic polynomial-time (PPT) adversary has success probability greater than ε to distinguish E(0, pk) and E(1, pk), where ε is negligible in k.

Matrix multiplicative perturbation

The most common method of data perturbation is that of additive perturbations. However, matrix multiplicative perturbations can also be used to good effect for privacy-preserving data mining. The data owner replaces the original data X with $Y = MX$ where M is an $n' \times n$ matrix chosen to have certain useful properties. If M is orthogonal ($n' = n$ n and $T M M = I$), then the perturbation exactly preserves Euclidean distances, *i.e.*, for any columns $x1$,$x2$ in X ,their corresponding columns $y1$,$y2$ in Y satisfy $x1 - x2 = y1 - y2$. If each entry of M is generated independently from the same distribution with mean zero and variance σ 2 (n' not necessarily equal to n), then the perturbation approximately preserves Euclidean distances on expectation up to constant factor 2σ n' . If M is the product of a discrete cosine transformation matrix and a truncated perturbation matrix, then the perturbation approximately preserves Euclidean distances.

HOMOMORPHIC RSA

Given RSA key pair (d, e)

- **$\varepsilon(x_1) = x_1^e$**
- **$\varepsilon(x_1x_2) = (x_1x_2)^e$**
- **$\varepsilon(x_1)\varepsilon(x_2) = (x_1^e)(x_2^e) = (x_1x_2)^e = \varepsilon(x_1x_2)$**

FULLY HOMOMORPHIC ENCRYPTION WITHOUT SQUASHING

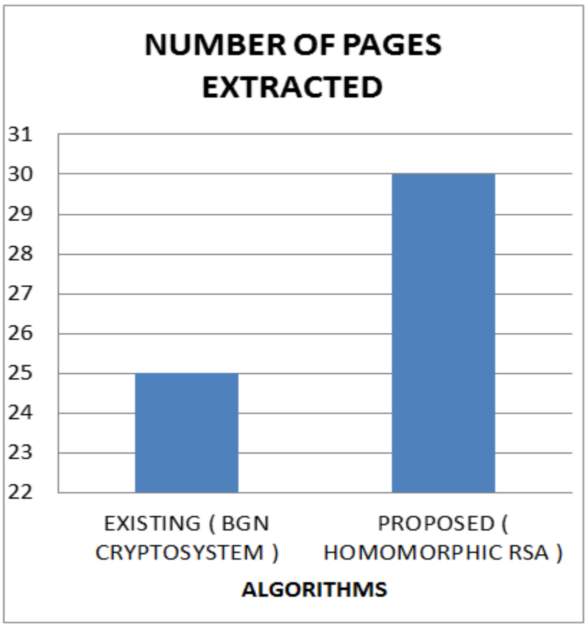
In 2009, Craig Gentry presented the first fully homomorphic encryption. This technique allowed one to compute arbitrary functions over encrypted data without the use of a decryption key. In his method, Gentry first constructs a somewhat homomorphic encryption, then compresses the decryption circuit to a more uncomplicated form. It is then bootstrapped to obtain a fully homomorphic encryption procedure. In 2011 Craig Gentry and Shai Halevi devised an advanced approach that consisted of a fusion of SWHE and another type of encryption called multiplicatively homomorphic encryption (MHE). This novel process eliminated the need for the compression step Gentry originally proposed in his dissertation. In this method, Gentry and Halevi devised a system to condense the FHE ciphertext into a single ciphertext whose security was superior. In ring theory or abstract algebra, a ring homomorphism is a function between two rings which preserves the operations of addition and multiplication. In their studies, they analyzed how to transform cipher texts over a big ring into small-ring ciphertexts that encrypt the same data. This procedure is known as ring switching. In their proposed method, they used a polynomial composition technique that splits a high degree polynomial into several lower degree polynomials. The idea behind this procedure was that the plaintext encrypted in the original large-ring packed ciphertext would be recovered as a simple linear function of the plaintexts encrypted in the smaller-ring ciphertexts. By now transferring smaller sizes of ciphertext instead of larger ones, the efficiency of the process would improve.

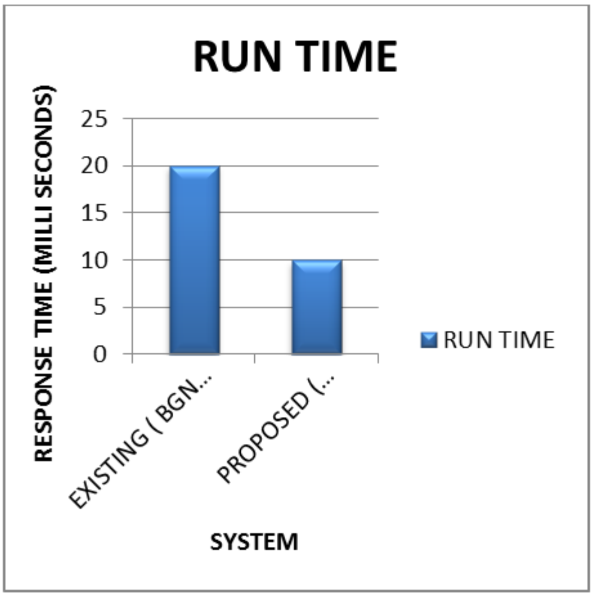
NOISE REDUCTION

There are some inevitable issues that arise when encrypting and decrypting data. In particular, the issue with noise reduction continues to be problematic during this process. Each conversion from plaintext to ciphertext has some noise associated with it. This noise continues to enlarge as one adds and multiplies ciphertexts, and as a result, the final ciphertext becomes indecipherable. A remedy to this concern was proposed by Gentry in his 2009 doctoral dissertation. He observed that if a noisy ciphertext could be decrypted and then re-encrypted, it would be restored with reduced noise. The problem was that this decryption would require a secret key, which was not available. His solution was to run the ciphertext through the decryption algorithm, but with an encrypted version of the decryption key. This resulted in a new ciphertext that contained lower noise and was as secure as the original ciphertext. this vice versa. Also included in his system was an evaluate function. This evaluate function can be thought of as a complete computer embedded into a cryptosystem. Its main purpose was to allow any computation to be performed on the ciphertext. This computation can be represented as a circuit or network where input signals traverse a series of boolean or logic gates. Because this evaluate function must be able to calculate any function, the circuit representing this function should be allowed to expand to any depth. The issue Gentry encountered, hence the motivation behind his work, was that after data has been encrypted, it contained a significant amount of noise. Gentry observed that if the noise continued to magnify, it would eventually overwhelm the signal. Because of this, the number of operations performed on the data would have to be restricted or inaccuracies would accrue. If the number of operations is limited then consequently, the circuit depth must also be limited. As a result, his system would not have been a fully homomorphic one but instead a somewhat homomorphic system because the number of operations performed on it is confined.

IV. RESULTS AND DISCUSSION

	EXISTING	PROPOSED
Number of Pages Extracted	25	30
Run Time	20	10





V. CONCLUSION

The proposed paper ensures that the user's online behavior data is completely privacy preserved. By Using asymmetric key encryption concept like RSA algorithm and the differential privacy makes sure that no one can assume the user's behaviors. The limitation in the number of queries answered will be increased considerably. In future, we can use images instead of noise for encrypting the user's data. In future, this method can use images instead of noise for encrypting the users' data. The limitation in the number of queries answered will be increased considerably. Other Homomorphic encryption methods can be tried to preserve the user data. Blowfish encryption, Honey encryption and AES encryption are alternate methods that can be used to preserve the online user data. In the future, the number of products can be extended to a larger dataset. The number of users also can be extended. Minimizing the information leakage during the computation and communication can be done. Another future work is to design privacy preserving data releasing protocols such that certain functions can be evaluated correctly while certain functional privacy can be protected.

VI. REFERENCES

[1]. Jianwei Qian, Fudong Qiu, Student Member, IEEE, Fan Wu, Member, IEEE, Na Ruan, Member,IEEE, Guihai Chen, Member, IEEE, and Shaojie Tang, Member, IEEE, "privacy preserving selective aggregation using onlune user behavior data",2016.

[2]. V. Rastogi and S. Nath, "Differentially private aggregation of distributed time-series with transformation and encryption," in Proceedings of the ACM International Conference on Management of Data (SIGMOD), 2010, pp. 735-746.

[3]. B. Applebaum, H. Ringberg, M. J. Freedman, M. Caesar, and J. Rexford, "Collaborative, privacy-preserving data aggregation at scale," in Proceedings of the 10th Privacy Enhancing Technologies Symposium (PETS), 2010, pp. 56-74.

[4]. R. E. Bucklin and C. Sismeiro, "Click here for internet insight: Advances in click stream data analysis in marketing," Journal of Interactive Marketing, vol. 23, no. 1, pp. 35-48, 2009.

[5]. H. Chen, R. H. Chiang, and V. C. Storey, "Business intelligence and analytics: From big data to big impact." MIS quarterly, vol. 36, no. 4, pp. 1165-1188, 2012.

[6]. Nathan Dowlin, Ran Gilad-Bachrach, Kim Laine , Kristin Lauter, Michael Naehrig, John Wernsin, "CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy", 33 rd International Conference on Machine Learning, New York, NY, USA, 2016. Volume 48, pp: 1 - 10.

[7]. Ihsan Jabbar, Saad Najim, "Using Fully Homomorphic Encryption to Secure Cloud Computing" Internet of Things and Cloud Computing, Volume 4, Issue 2, April 2016, Pages: 13-18.

[8]. Jung Hee Cheon, Andrey Kim, Miran Kim and Yongsoo Song, "Homomorphic Encryption for Arithmetic of Approximate Numbers", International Conference on the Theory and Application of Cryptology and Information Security ASIACRYPT 2017, pp 409-437.

[9]. Sungwook Kimy, Jinsu Kimz, Dongyoung Koox, Yuna Kim, Hyunsoo Yoonk and Junbum Shin, "Efficient Privacy-Preserving Matrix Factorization via Fully Homomorphic Encryption" 11th ACM on Asia Conference on Computer and Communications Security, May 2016, pp: 617-628.

[10]. Ana Costache and Nigel P. Smart,"Which Ring Based Somewhat Homomorphic Encryption Scheme is Best" CT-RSA 2016: Topics in Cryptology, 2016, pp: 325-340.

[11]. Y. Aono, T. Hayashi, L. T. Phong, and L. Wang, "Efficient homomorphic encryption with key rotation and security update," IEICE Trans. Inf. Syst., vol. E101-A, no. 1, pp. 39-50, 2018.

[12]. W. Banaszczyk, "New bounds in some transference theorems in the geometry of numbers," Math. Annalen, vol. 296, no. 1, pp. 625-635, 1993.

[13]. W. Banaszczyk, "Inequalities for convex bodies and polar reciprocal lattices in Rn ," Discrete

Comput. Geometry, vol. 13, no. 1, pp. 217-231, 1995.

[14]. K. Bonawitz et al., "Practical secure aggregation for privacy preserving machine learning," in Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS), 2017, pp. 1175-1191.

[15]. I. Chillotti, N. Gama, M. Georgieva, and M. E. Izabachène "Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds," in Proc. Adv. Cryptol.-ASIACRYPT, 2016, pp. 3-33.

[16]. J. Dean et al., "Large scale distributed deep networks," in Proc. 26th Annu. Conf. Neural Inf. Process. Syst., 2012, pp. 1232-1240.

[17]. J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," J. Mach. Learn. Res., vol. 12, pp. 2121-2159, Feb. 2011.

[18]. R. Gilad-Bachrach et al., "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy," in Proc. 33nd Int. Conf. Mach. Learn. (ICML), vol. 48. 2016, pp. 201-210.

[19]. O. Goldreich, Foundations of Cryptography: Basic Applications. vol.2. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[20]. B. Hitaj, G. Ateniese, and F. Pérez-Cruz, "Deep models under the GAN: Information leakage from collaborative deep learning," in Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS), 2017, pp. 603 - 618.