International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

New Secure Data Protection Scheme for Data Privacy Protection Based on Complete Homomorphic Encryption in Link with Logistic Regression

Mr. Thirupathi Nanuvala[1], Dr. Bodla Kishor[2], Mr. Manchikatla Srikanth[3], Dr. Syed Shabbeer Ahmad[4]

Submitted: 05/05/2024 **Revised**: 18/06/2024 **Accepted**: 25/06/2024

Abstract: More and more, healthcare facilities and academic institutions are relying on technological solutions to securely share patient information. Technical solutions such as distributed ledger technology and homomorphic encryption are available. It is possible to compute data without ever needing to decode it using homomorphic encryption. Cloud data is still a target for attackers due to the inherent instability and fast progress of technology. Consequently, homomorphic encryption offers a practical way to test the durability of collections of sensitive patient data kept in different regions. As a result, a homomorphic encryption method based on matrix transformations was used to shift, rotate, and transpose each letter in the converted Binary ASCII value of the original text. Symmetric cryptography encrypts and decrypts using the same secret key. One advantage of symmetric encryption is the "avalanche efect," which occurs when two different keys generate different cypher texts for the same communication. The key's varied circumstances are the source of this effect. The cryptanalysis of the proposed technique shows that it is more secure than current encryption approaches against a wide variety of attacks. a way that malicious actors' statistical analyses can't simply deduce the plaintext.

Keywords: *homomorphic, malicious, cryptography, instability, ASCII,*

1 Introduction:

Over the last several years, there has been a tremendous advancement in information technology. Cloud computing has seen an increase in applications, and homomorphic encryption methods are used extensively in cloud storage and cloud computing [1]. The need to protect users' personal information has grown in tandem with the popularity of cloud computing and must be handled as an urgent matter of security. However, there isn't a trustworthy cloud centre for the public key cryptosystem-based homomorphic encryption method, and the method's more complicated calculation process means it can't handle lots of users or lots of calculations [2-3]. Consequently, both user and data privacy are safeguarded throughout the aggregation process by using the homomorphic encryption technique. The aggregated result is sent to the cloud centre, and no personally identifiable information about users is revealed, guaranteeing data security. When compared to the RSA encryption technique, the homomorphic encryption approach clearly excels in processing speed and data privacy protection, demonstrating its ability to securely safeguard users' data.

1.1 Motivation

Cloud computing's primary issue is security. One of the major problems is the data transmission to third-party service providers for processing and automation. The private and sensitive information of its customers is valuable to every company or organisation. Research, new marketing campaigns, laws, and creative product launches are all made possible with data, which is why all organizations—public, corporate, healthcare, and academic alike—need it. Acumen Research and Consulting predicts that the healthcare cloud computing industry will reach $40 billion by 2026, therefore protecting patients' personal health information is our top priority [4]. The healthcare business may benefit from cloud computing in two ways: increased productivity and reduced expenses. Despite its speed, protecting sensitive healthcare information is essential for boosting patient trust and driving economic growth. Saving money and making healthcare more efficient are two goals of digitising patient medical records. However, patient records include a great deal of personal information. Consequently, patients must be able to promptly and reliably provide access to their personal information to several medical associations in a safe environment. It is critical to study the healthcare industry's usage of homomorphic encryption and evaluate many

[1]*Assistant Professor, Department of CSE V.N.R Vignana Jyothi Institute of Engineering and Technology, Hyderabad, India*
thirupathi_n@vnrvjiet.in
[2]*Associate Professor, Department of CSE CMR Engineering College, Hyderabad, India*
trml.kishore@gmail.com
[3]*Research Scholar, Department of CSE, UCE, Osmania University,Hyderbad,India Assistant Professor, Department of CSE, V.N.R Vignana Jyothi Institute of Engineering and Technology, Hyderabad, India*
srikanth_m@vnrvjiet.in
[4]*Professor, Department of CSE, MJCET, Hyderabad, India*
shabbeer.ahmad@mjcollege.ac.in

homomorphic methods to ensure patient privacy during data searches and disease prediction.
Here is how the remainder of the article is structured: Methods, planning, criteria for inclusion and exclusion, and research questions accompanied by rationale will be covered in the following section. Partially and moderately homomorphic approaches, as well as completely homomorphic methods, are compared and examined in the next section. We classified fully homomorphic encryption methods into four categories and compared the most important tactics from each. Homomorphic encryption's applications in healthcare will be covered in the next section. Homomorphic encryption methods were evaluated for security based on computation and transmission costs.

2 Related Work

To facilitate the development of web applications and to serve as a reliable firewall between the client and the server, Martin Johns et al. [13] assembled Crypto Membranes, a set of native client-side components. On the client side, data encryption protects against untrustworthy third parties. While continuing to be totally compatible with the approaches employed in current client-side programming. To provide consumers a real experience, he also shows how Crypto Membranes may be integrated with existing web browsers by using a conventional browser during the extension's transition period.

The most current assaults on efficient encrypted cloud data search enabled by SSE or OPE/ORE were outlined in an orderly fashion by Yao et al. [14]. They have classified the rival model using many criteria in order to be more specific. Under ten different sorts of attackers, they examined the current attacks on OPE/ORE and SSE and the security holes that these frameworks allow.
A pattern outlining its design, identifying its issues, and proposing remedies was published by Eduardo B. Fernandez et al. [15]. The pattern safeguards data assets and communication paths to avoid attacks on IoT devices and to make security management easier. A few examples of security methods include auditing and security loggers, firewalls and intrusion detection systems, secure channels, and permission-based authentication. The team is building a library of patterns to ensure the safety of the Internet of Things ecosystem, and this is one of their contributions.

One method that was suggested by Shruthi Ramesh et al. [16] is known as proxy re-ciphering as a service. Encrypted data remains private even after a device-key breach because to the integration of dispersed servers, chameleon hash functions, FHE, and secret sharing, which form a robust and durable system. In order to evaluate the system, they set up a testbed and tracked the latencies using real ECG records taken from the TELE ECG database.
Consistent with what has been found by Kanchanadevi and colleagues [17] Protecting sensitive information in a hybrid cloud requires encryption. There are a lot of encryption methods at our disposal, but they all bring up valid points about the safety of our data. The attribute-based encryption system that supports dynamic attributes (ABE-DAS) has overcome these problems. Combining a dynamic attribute strategy with an attribute-based encryption method improves the security of data stored in a hybrid cloud. Whether your data is structured or not, the ABE-DAS encryption technique can handle it when you use an attribute approach. It turns regular text into cypher text by combining static and dynamic properties.

K. Naregal et al. [18] state that the dramatic increase in cloud computing and IoT devices has led to easy, efficient, and secure data access. It has been concluded that a method based on lightweight attributes is necessary for the cloud-based Internet of Things (IoT).

Simple methods may be sufficient to meet the proposed and achieved communication security needs of Zeesha Mishra et al. [19]. Utilising and modelling the best lightweight cyphers is essential. It is necessary to mimic the design in order to include the cypher into hardware that may be used to track different parameters. All three cyphers—TEA, XTEA, and XXTEA—were used to accomplish the desired objective. Developing, implementing, and optimising these cyphers required the utilisation of FPGA and ASIC technology. There are a lot of things that have been considered, such as block sizes, implementation rounds, and important scheduling components.

To encrypt RGB images, Dina Ibrahim et al. [20] came up with a novel approach. Each pixel in an RGB picture is encrypted using a technique that involves 16 rounds of DNA encoding, transpositions, replacements, and chaotic systems. Round keys are generated at random using a logistic chaotic function, which also randomly creates a 16x16 nonlinear matrix. Over the course of several rounds, these keys are used in conjunction with the DNA Playfair matrix to alter specific pixels. Data encryption and decryption using the proposed approach is faster and more secure than existing methods, according to experiments. The numerical measurements show that the proposed method may protect reference evaluation values against statistical and differential attacks while keeping them intact.

A machine learning strategy that addresses data privacy protection and multi-party cooperative learning,

federated learning was introduced by Google in 2016 [6]. A growing number of institutions and businesses are beginning to embrace federated learning, a cooperative machine learning strategy involving several parties [7, 8]. Federated learning initially aimed to assist Android users in resolving the issue of model localisation. Federated learning might be used to other areas of machine learning as well. It was stated in 2019 [13] that Google scientists used Tensorflow to build a scalable production system for mobile device-related collaborative learning. Furthermore, for 2019, additional relevant assignments have been suggested. Without providing raw data to a centralised node, Wang highlighted the issue of learning model parameters when data is spread across numerous edge nodes [12]. Additionally, federated transfer learning has been the subject of some research. It is possible to apply the architecture described in [11] with various types of safe multi-party machine learning because of its adaptability. Concerning efficiency, [7] presented a framework called SecureBoost that achieved almost same accuracy as the five privacy protection solutions.

Federated learning has found widespread application across many domains. To illustrate, one example is Google's Gboard technology, which enables keyboard input prediction and enhances input efficiency while protecting user privacy [8,9]. The ability to securely manage sensitive patient data makes federated learning an ideal tool for the medical industry [10,11]. Federated learning also has applications in recommendation systems [23] and natural language processing [12].

Furthermore, there has been a great deal of recent substantial work on machine learning to safeguard privacy. Using SMC to mitigate differential privacy-induced noise was in line with the recommendations made by Zhou et al.[24,25] for protecting machine learning privacy. In 2020, Zhang et al. suggested an optimization-based batchcrypt method using the FATE framework [26]. By minimising the amount of mathematics required, encryption and decryption speed is improved when a batch of quantised gradients is encrypted as a long integer all at once. To improve Bayesian machine learning, Wei Ou et al. developed a vertical federated learning system that uses homomorphic encryption. Their model outperforms a single union server training model by 90% [27].

3 METHODOLOGY:

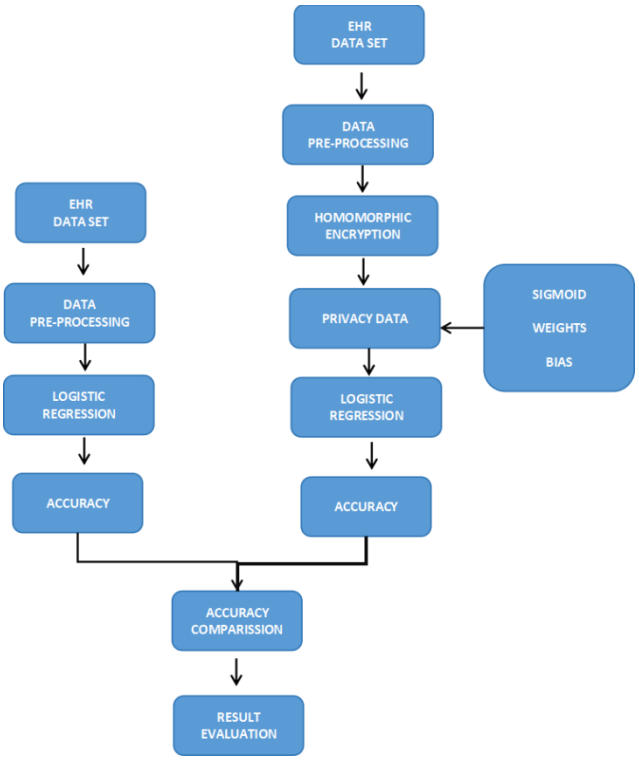

Fig 1: Workflow model

3.1 HOMOMORPHIC ENCRYPTION:

An example of a public key cryptography technique is homotopy encryption (HE). After the user generates a pair of public and private keys, she encrypts her data using the public key before sending it to an outside party for processing. Due to the fact that encryption and decryption are synonymous, the user may see the outcome of the calculation performed on her data by decrypting it using her own key. So long as the user never gives out their personal details, they may see the results of the calculation in this way. Once encryption is finished, almost all computing may be outsourced to the server utilising HE. Homomorphic properties enable the user to decode the output and do calculations on encrypted data securely, without compromising privacy. It may take more time to do certain computations with HE since it is limited to addition and multiplication and cannot efficiently handle arbitrary big multiplications.

Processing data in its unencrypted form is the norm due to the fact that encryption techniques prohibit handling encrypted data. Homomorphic encryption, in contrast, lets the user decrypt findings after computing on encrypted material. Not only does homomorphic encryption permit processing of encrypted data, but it also guarantees privacy continuously. The three algorithms that make up traditional public-key encryption are key creation, encryption, and decryption. On the other hand, homomorphic encryption is an essential characteristic.

3.2 FULLY HOMOMORPHIC ENCRYPTION (FHE):

Classical encryption allows you to encrypt data in an encrypted "box," or safe, and then secure it with a key. Important data operations, including searches or computations, need the key to "unlock" or decrypt the data. Encryption makes your data more susceptible to hackers, who might potentially steal your personal information. For its part, FHE enhances encryption by removing the need to disclose the underlying data in order to work directly with encrypted data. Envision your data kept in a locked "box." Using FHE, you can manipulate the data even when it's still in the box. Data privacy is maintained at every stage by further encrypting the output of these procedures. When it comes to safe data exchange and analysis, Fully Homomorphic Encryption provides a revolutionary answer to the healthcare industry's data privacy challenges. Medical organisations may work together or talk about research, diagnoses, and treatment plans securely using FHE to do computations on encrypted data, eliminating the danger of exposing sensitive patient information. With FHE, you can do any maths you want on encrypted data. The fact that all functions may be defined as Boolean circuits means that any encryption technique capable of adding and multiplying can potentially evaluate any function. Levelled homomorphic encryption (LHE) are methods that can execute homomorphic operations over a circuit of a certain depth. The practical range of functions that can be calculated with FHE/LHE is limited by factors such as ciphertext noise building, key sizes, ciphertext sizes, execution time, and circuit depth.

3.3 IMPLEMENTING FULLY HOMOMORPHIC ENCRYPTION:

New, open-source FHE libraries and apps have emerged as a result of recent FHE research. The most current community standards for homomorphic encryption and state-of-the-art FHE research are continually being considered as these libraries evolve [9]. Even while FHE is becoming more effective, non-experts still struggle to use it in practical settings. A great deal of programming and cryptographic knowledge is needed for the custom implementations that are necessary. Some examples include the difficulty of translating calculations from plaintext to ciphertext, the need to manually configure security settings, and the careful management of noise to ensure accurate decryption [7]. Several preliminary compilers and tools have been released to aid in making FHE more accessible and usable.

The PYthon For Homomorphic Encryption Libraries (Tenseal & Pyfhel) use PALISADE, HElib, and SEAL as backends to provide homomorphic encryption methods in Cython [3]. The Python library PySyft offers many methods for ensuring privacy, including homomorphic encryption [8]. Encryption that is semi-homomorphic or somewhat-homomorphic is available in some libraries. For Python 3, the partly homomorphic Paillier method may be used with the help of the PythonPaillier package [9]. To implement the moderately homomorphic DHS method on a GPU, one needs the CUDA Homomorphic Encryption Library (cuHE) [5]. The Awesome Homomorphic Encryption website is a good place to find newly developed FHE libraries.

Several different frameworks fall under the umbrella term "private-preserving machine learning," and they may all be used to train and categorise sensitive data. These techniques could include a lot of people, including the owner(s) of the data, the model(s), and the server(s) on the cloud. One option is to have the client's encrypted data processed by a server in the cloud. The client receives the result (encrypted) and uses it for private evaluation after decrypting it. Implementing privacy-preserving machine learning has many potential applications, including personalised medicine, diagnostic tools, and DNA sequence analysis [11, 12].

3.4 LOGISTIC REGRESSION

The posterior probability of the classes is expressed using linear functions in logistic regression methods. The model has the following form when given a set of points {1, x1, x2,..., xn}, where xi is a binary variable and {−1, 1} are binary classes.

$$\Pr(Y=-1|\ X=x)=\frac{1}{1+e^{-\beta T}x}$$

where Pr(Y = 1|X = x) = 1 − Pr(Y = −1|X = x) the weight vector is denoted by β.Being acquainted with the parameter set β is a precondition for training a logistic regression model. Make a Logistic Regression Model. Finding the parameters of a logistic regression model is possible using the maximum likelihood estimation method. If pyi is assumed (xi ; β) = Pr(Y = yi | X = x). If we have N observations with binary class labels, we may calculate their log-likelihood by

$$\ell(\beta)=\sum_{i=1}^{N}\log py_i(x_i;\beta)=\sum_{i=1}^{N}(y_i\beta^T x_i-\log(1+e^{yi}x_i))$$

in the set {−1, 1} where yi is. The log-likelihood may be optimised in several ways. Newton's technique iteratively updates to estimate the zeros of the gradient of ℑ.

$$\beta_{k+1}=\beta_k+\gamma k\sum_{i=1}^{N}(1-\sigma(\beta^T x))y_i x_i$$

where σ is the sigmoid function, σ (x) = 1/(1 + exp(x)).

A change to the optimisation function that is HE-friendly is necessary to train logistic regression techniques utilising encrypted input. Due to the price of homomorphic computing, the training approach should need the fewest repetitions for convergence. Because it converges more quickly than conventional gradient descent, Kim et al.'s efficient technique learns its parameter set using Nesterov's accelerated gradient [16].

A neural network's architecture consists of several layers, the most common of which are the activation, convolution, pooling, and dense (or completely coupled) layers. Direct usage of homomorphic encryption is not possible since certain of the functions of these layers are not polynomial. Picking the right precision bits for inputs and network weights and utilising polynomial functions to approximate non-linear layers is essential for keeping network accuracy low during homomorphic assessment. Nonlinear layers have been approximated using a wide variety of methods in homomorphic evaluation.

Convolutional layers retrieve characteristics using discrete convolution. This takes a vector g as input, uses it to calculate the weighted sums of a vector of input values f, and then passes that vector on to the next network layer. In the context of image processing, if f and g are matrices, then

$$f * g = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f(i, j) g(m-i, n-j)$$

Picture classification networks often include convolutional layers, where f is a picture subregion centred at index m and g is a kernel matrix [12]. It is possible to encrypt the kernel g weights after training is complete. The use of FHE allows for the computation of discrete convolution as a polynomial function.

several levels of activation! To make the neural network model non-linear, activation layers are used. Activation layers often follow convolution layers. A few popular activation functions include the sigmoid, hyperbolic tangent, and REctified Linear Unit (ReLU) functions. Max(0, x) is the ReLU function, and the sigmoid is given by

$$\phi(z) = \frac{1}{1+\exp(-z)}$$

Special issues arise when training privacy-preserving models on encrypted data. It is common practice to evaluate the network's accuracy while training it in order to include training improvements when dealing with difficult classification problems. Having said that, the network must demonstrate strong performance during its first training session using encrypted data [25]. In addition, training on homomorphically encrypted data requires assessing multiple homomorphic methods, which may be a major processing burden.

4 Experiments & Results

```
[ ] def encrypted_evaluation(model, enc_x_test, y_test):
        t_start = time()

        correct = 0
        for enc_x, y in zip(enc_x_test, y_test):
            # encrypted evaluation
            enc_out = model(enc_x)
            # plain comparison
            out = enc_out.decrypt()
            out = torch.tensor(out)
            out = torch.sigmoid(out)
            if torch.abs(out - y) < 0.5:
                correct += 1

        t_end = time()
        print(f"Evaluated test_set of {len(x_test)} entries in {int(t_end - t_start)} seconds")
        print(f"Accuracy: {correct}/{len(x_test)} = {correct / len(x_test)}")
        return correct / len(x_test)


    encrypted_accuracy = encrypted_evaluation(eelr, enc_x_test, y_test)
    diff_accuracy = plain_accuracy - encrypted_accuracy
    print(f"Difference between plain and encrypted accuracies: {diff_accuracy}")
    if diff_accuracy < 0:
        print("Oh! We got a better accuracy on the encrypted test-set! The noise was on our side...")
```

```
Evaluated test_set of 334 entries in 1 seconds
Accuracy: 231/334 = 0.6916167664670658
Difference between plain and encrypted accuracies: 0.011976063251495361
```

The models' assessment times varied by 0.19 seconds when comparing encrypted and non-encrypted data. Because of this, the encypted works outperform the palin-data.

Training an Encrypted Logistic Regression Model on Encrypted Data

In order to train the encrypted logistic regression model, we will need to create a model similar to PyTorch that can forward and backpropagate encrypted input. This will allow us to update the weights using the encrypted data. Further information on the training may be found here.

Loss Function

Specifically, we use the regularised binary cross entropy loss function (the rationale for regularisation will be discussed at a later point). The i'th predicted label is denoted by y(i), and Θ represents the n-sized weight vector. Y^(i) stands for the output of the i'th logistic regression model.

Loss(θ)=−1m∑i=1m[y(i)log(y^(i))+(1−y(i))log(1−y^(i))] +λ2m∑j=1nθ2j

Parameters Update

Typically, while updating a parameter, the following steps are taken, with x(i) standing for the i-th input data point:
The equation θj=θj−α[1m∑i=1m(y^(i)−y(i))x(i)+λmθj] can be paraphrased as; it is a functional equation.

We choose to utilise λm=0.05 and α=1 to decrease a multiplication due to the homomorphic encryption requirement. As a consequence, the modification that follows the

rule:θj=θj−[1m∑i=1m(y^(i)−y(i))x(i)+0.05θj]

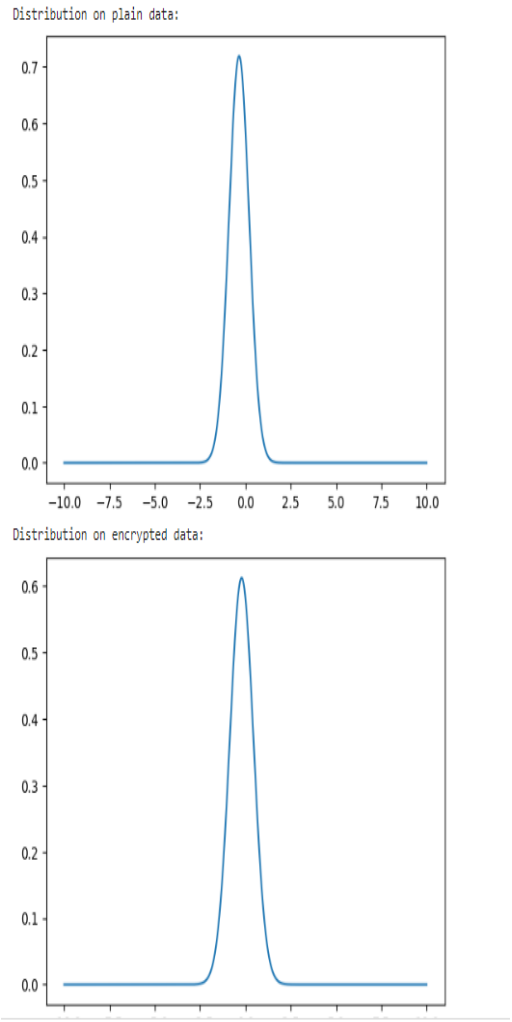

Fig 2: Data distribution on both data

```
Accuracy at epoch #0 is 0.523952066898346
Accuracy at epoch #1 is 0.6856287717819214
Accuracy at epoch #2 is 0.697604775428772
Accuracy at epoch #3 is 0.6916167736053467
Accuracy at epoch #4 is 0.6856287717819214
Accuracy at epoch #5 is 0.6856287717819214

Average time per epoch: 107 seconds
Final accuracy is 0.6856287717819214
Difference between plain and encrypted accuracies: 0.017964065074920654
```

Fig 3: The data accuracy is 68%, which is higher than previous models.

There has long been a consensus among statisticians that PHI should not be stored in statistical databases. Instead of disturbing the data using data masking techniques to provide basic summary statistics, official statistics efforts have primarily focused on accurate statistical inference for more complex statistical models.

Here we take a look at the issue of multiple regression computation in a scenario where the data are spread out across several sources, and none of them are willing to share their data. This issue arises, for instance, when data comprises billing and record information from health insurance plans and the parties involved are health insurance companies. Despite the fact that the data cannot be delivered due to legal obstacles, a regression using the combined sets of data from both parties may have better features than using each (incomplete) set. Such issues emerge whenever two or more parties conduct repeated polls of the same group of people. Following that, they want to do regressions using the aggregated survey variables; nevertheless, they are hesitant to provide the data.

This research delves into a distinct but connected privacy protection issue within the framework of regression analysis, which pertains to statistical computations conducted on several datasets. In this case, the two primary parties concerned with privacy are the database owners and the people whose data is stored in various databases. With respect to the former, the privacy safeguards afforded to data inside individual sources are insufficient to cover the connected individual files when data is merged across sources. Database owners may not want to or be able to share their data directly with other parties, even if safeguarding individual privacy is not a problem.

A technique for secure matrix products provides a comparable foundation for the secure regression due process. On the other hand, after the writers have safely computed the whole data covariance matrix and disseminated it to all users (along with the response vector), they cease writing. This reduces data privacy but makes the protocol computationally interesting by allowing parties to calculate a broad range of diagnostics locally. The disclosure of more personal information than is absolutely necessary occurs due to the fact that the data covariance matrix is not required in order to produce the coefficient vector.

The majority of prior work on privacy-preserving data mining has been on situations where the parties' data distributions are predictable. The two most typical kinds are "vertical partitioning," in which each side has a subset of the attributes, and "horizontal partitioning," in which each side has a subset of the instances. We provide a procedure that may be used in any case where the parties involved have interdependent database components, independent of the partitioning method. When data warehouses are involved, this might happen.

We construct a computation protocol. The protocol is a

series of steps that allow users to connect with one other and execute calculations locally. To implement the concept of security, we want to use a "semi-honest" method of cryptography. The underlying assumption of this security model is that not only will all participants adhere to the rules and disclose their actual input values, but they will also be interested in discovering the hidden inputs of their peers. If the messages sent between the parties while the protocol is running do not reveal any information about the private inputs that each party possesses, then the protocol is considered secure. When it comes to security, the only thing needed to "simulate" a party's message transcript is their knowledge of the protocol's output and input. To do this, a formal specification of a probabilistic polynomial time approach (the simulator) is necessary. A party, their output, and a randomly selected seed are the three inputs. After that, it creates a transcript of the message that is computationally identical to the one that would be created during the execution of the protocol.

5. Conclusion

From Gentry's original proposal to the efficient implementations now on the market, research has led to a rapid improvement in completely homomorphic encryption. Several open-source libraries that support FHE are now accessible. A thorough familiarity with the theory behind the scheme or schemes is usually necessary for implementing these FHE libraries. The present efforts will result in the availability of more advanced, user-friendly technologies in the future, but even with these implementations, non-experts still have a hard time navigating. This would open the door for researchers from many walks of life to use FHE. These enhancements would have the greatest impact on the healthcare industry, which handles a great volume of sensitive patient information. A doctor could be able to anonymously access a prediction model using privacy-preserving classification methods, all without disclosing any patient health information. Among FHE's many uses in statistics and ML is its incorporation into private logistic regression evaluations.There will be more uses for FHE as it becomes faster and more useful.

References:

[1] 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). Office Journal of the European Union L 119 (April 2016), 1ś88.

[2] 2018. The PALISADE Lattice Cryptography Library. https://palisade-crypto.org/software-library/.

[3] 2018. PYthon For Homomorphic Encryption Libraries (Pyfhel). https://github.com/ibarrond/Pyfhel.

[4] 2018. RAMPARTS: RApid Machine-learning Processing Applications and Reconfigurable Targeting of Security.https://galois.com/project/ramparts/

[5] 2019. Lattigo 1.3.0. Online: http://github.com/ldsec/lattigo. EPFL-LDS.

[6] Abbas Acar, Hidayet Aksu, A. Selcuk Uluagac, and Mauro Conti. 2018. A Survey on Homomorphic Encryption Schemes: Theory and Implementation. ACM Comput. Surv. 51, 4 (July 2018), 79:1ś79:35.

[7] Rakesh Agrawal and Ramakrishnan Srikant. 2000. Privacy-preserving Data Mining. SIGMOD Rec. 29, 2 (May 2000), 439ś450.

[8] C. Aguilar-Melchor, S. Fau, C. Fontaine, G. Gogniat, and R. Sirdey. 2013. Recent Advances in Homomorphic Encryption: A Possible Future for Signal Processing in the Encrypted Domain. IEEE Signal Processing Magazine 30, 2 (March 2013), 108ś117.

[9] Miklós Ajtai and Cynthia Dwork. 1997. A Public-key Cryptosystem with Worst-case/Average-case Equivalence. In Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing (STOC '97). ACM, New York, NY, USA, 284ś293.

[10] Hany Alashwal, Mohamed El Halaby, Jacob J. Crouse, Areeg Abdalla, and Ahmed A. Moustafa. [n.d.]. The Application of Unsupervised Clustering Methods to Alzheimer's Disease. 13 ([n. d.]).

[11] Martin Albrecht, Shi Bai, and Léo Ducas. 2016. A Subfield Lattice Attack on Overstretched NTRU Assumptions. In Proceedings, Part I, of the 36th Annual International Cryptology Conference on Advances in Cryptology Đ CRYPTO 2016 - Volume 9814. Springer-Verlag, Berlin, Heidelberg, 153ś178.

[12] Martin Albrecht, Melissa Chase, Hao Chen, Jintai Ding, Shafi Goldwasser, Sergey Gorbunov, Shai Halevi, Jeffrey Hoffstein, Kim Laine, Kristin Lauter, Satya Lokam, Daniele Micciancio, Dustin Moody, Travis Morrison, Amit Sahai, and Vinod Vaikuntanathan. 2018. Homomorphic Encryption Security Standard. Technical Report. HomomorphicEncryption.org, Toronto, Canada.

[13] Jacob Alperin-Sheriff and Chris Peikert. 2014. Faster Bootstrapping with Polynomial Error. In Advances in Cryptology - CRYPTO 2014 (Lecture Notes in Computer Science). Springer, Berlin, Heidelberg, 297ś314.

[14] R. Altman, E. Asch, D. Bloch, G. Bole, D. Borenstein, K. Brandt, W. Christy, T. D. Cooke, R. Greenwald, M. Hochberg,D. Howell, D. Kaplan, W. Koopman, S. Longley III, H. Mankin, D. J. McShane, T. Medsger Jr., R. Meenan, W. Mikkelsen, R. Moskowitz, W. Murphy, B. Rothschild, M. Segal, L. Sokoloff, and F. Wolfe. 1986. Development of criteria for the classification and reporting of osteoarthritis: Classification of osteoarthritis of the knee. Arthritis & Rheumatism 29, 8(Aug. 1986), 1039ś1049.

[15] Yoshinori Aono, Takuya Hayashi, Le Trieu Phong, and Lihua Wang. 2016. Scalable and Secure Logistic Regression via Homomorphic Encryption. In Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy (CODASPY '16). ACM, New York, NY, USA, 142ś144.

[16] Frederik Armknecht, Colin Boyd, Christopher Carr, Kristian Gùsteen, Angela J''aschke, Christian A. Reuter, and Martin Strand. 2015. A Guide to Fully Homomorphic Encryption. Technical Report 1192.

[17] Frederik Armknecht, Stefan Katzenbeisser, and Andreas Peter. 2013. Group homomorphic encryption: characterizations, impossibility results, and applications. Designs, Codes and Cryptography 67, 2 (May 2013), 209ś232.

[18] Mikhail J. Atallah, Florian Kerschbaum, and Wenliang Du. [n.d.]. Secure and private sequence comparisons. In Proceeding of the ACM workshop on Privacy in the electronic society - WPES '03 (2003). ACM Press, 39.

[19] Erman Ayday, Jean Louis Raisaro, Jean-Pierre Hubaux, and Jacques Rougemont. [n.d.]. Protecting and evaluating genomic privacy in medical tests and personalized medicine. In Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society (2013-11-04) (WPES '13). Association for Computing Machinery, 95ś106.

[20] Md Momin Al Aziz, Md Nazmus Sadat, Dima Alhadidi, Shuang Wang, Xiaoqian Jiang, Cheryl L Brown, and Noman Mohammed. [n.d.]. Privacy-preserving techniques of genomic dataÐa survey. ([n. d.]).

[21] Jean-Claude Bajard, Julien Eynard, M. Anwar Hasan, and Vincent Zucca. 2017. A Full RNS Variant of FV Like Somewhat Homomorphic Encryption Schemes. In Selected Areas in Cryptography ś SAC 2016, Roberto Avanzi and Howard Heys (Eds.). Springer International Publishing, Cham, 423ś442.

[22] Jean-Claude Bajard, Julien Eynard, Paulo Martins, Leonel Sousa, and Vincent Zucca. 2019. An HPR variant of the FV scheme: Computationally Cheaper, Asymptotically Faster. Cryptology ePrint Archive, Report 2019/500.

[23] https://eprint.iacr.org/2019/500.

[24] Pierre Baldi, Roberta Baronio, Emiliano De Cristofaro, Paolo Gasti, and Gene Tsudik. [n.d.]. Countering GATTACA: Efficient and Secure Testing of Fully-sequenced Human Genomes. In Proceedings of the 18th ACM Conference on

[25] Computer and Communications Security (2011) (CCS '11). ACM, 691ś702.

[26] Josh Benaloh. 1987. Verifiable Secret-ballot Elections. Ph.D. Dissertation. New Haven, CT, USA. AAI8809191.

[27] Josh Benaloh. 1994. Dense Probabilistic Encryption. In In Proceedings of the Workshop on Selected Areas of Cryptography. 120ś128.

[28] Daniel Benarroch, Zvika Brakerski, and Tancrède Lepoint. 2017. FHE over the Integers: Decomposed and Batched in the Post-Quantum Regime. In Public-Key Cryptography - PKC 2017, Serge Fehr (Ed.). Vol. 10175. Springer Berlin Heidelberg, 271ś301.

[29] R. Bender and U. Grouven. 1997. Ordinal logistic regression in medical research. 31, 5 (Oct. 1997), 546ś551.

[30] [Jean-François Biasse and Luis Ruiz. 2015. FHEW with Efficient Multibit Bootstrapping. In Progress in Cryptology ś LATINCRYPT 2015 (Lecture Notes in Computer Science), Kristin Lauter and Francisco Rodríguez-Henríquez (Eds.).

[31] Springer International Publishing, 119ś135.

[32] Marcelo Blatt, Alexander Gusev, Yuriy Polyakov, Kurt Rohloff, and Vinod Vaikuntanathan. 2019. Optimized Homomorphic Encryption Solution for Secure Genome-Wide Association Studies. Cryptology ePrint Archive, Report 2019/223. https://eprint.iacr.org//2019//223.

[33] Fabian Boemer, Yixing Lao, Rosario Cammarota, and Casimir Wierzynski. [n.d.]. nGraph-HE: a graph compiler for deep learning on homomorphically encrypted data. In Proceedings of the 16th ACM International Conference on Computing Frontiers (2019-04-30) (CF '19). Association for Computing Machinery, 3ś13.

[34] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. 2005. Evaluating 2-DNF Formulas on Ciphertexts. In Proceedings of the Second International Conference on Theory of Cryptography (TCC'05). Springer-Verlag, Berlin, Heidelberg, 325ś341.