InfoCrypt: Real World Implementation of Machine Learning on Encrypted Data

[1] Piyush Agarwal, [2] Shalette Natasha D'Souza, [3] Sharanya Warrier, [4] Sourav Ray Chaudhuri
Department of Computer Science & Engineering
BMS Institute of Technology & Management
Bengaluru, Karnataka, India
Email : [1]1by16cs059@bmsit.in, [2]1by16cs074@bmsit.in, [3]1by16cs109@bmsit.in, [4]1by16cs081@bmsit.in

***Abstract*—In the modern age of cloud computing, the demand of data processing on cloud is increasing by the day. Companies are onboarding new technologies as fast as possible, but in this haste they've left a major part – data privacy – underdeveloped. This has resulted in the trade of personal data for amazing new features. Hence, there has been a huge increase in targeted or personalized advertisements, scams and frauds. People never question how scammers know whom or what category of people to phish and how it's done, how some companies know what people want to buy or what their buying capacity is when they've never visited their websites, how insurance companies know if people are eligible for their policies without any previous contact. If third party companies can't be trusted with user data, the responsibility falls upon the user to demand changes to protect their data. This project, titled 'InfoCrypt' is an implementation of partially homomorphic encryption schemes in a practical use case. In this project, we want to show that machine learning queries can be performed while retaining user privacy. Rather than following a mutual trust model with third party companies that is failing hard in modern times, we are placing user privacy in the hands of users themselves.**

***Index Terms*—Paillier, Cryptosystem, Machine Learning, Partially, Homomorphic, Encryption, InfoCrypt**

I. Introduction

At present, many companies have started using machine learning to provide better services to their users. Machine learning makes accurate results based on most relevant data it is trained on. The user in need to use the service needs to provide that data to the model via some API request. If the data is sensitive such as bank balance, annual salary etc. the companies can sell that data to other companies for profit which may make users prone to targeted ads such as insurance agencies, property brokers etc.

This can be easily avoided by using a Partially Homomorphic Encryption (PHE) system [1] [2]. Here we can encrypt the numbers such that some arithmetic operations can be done on the encrypted value. We can add and subtract to encrypted numbers, multiply or divide scalar numbers with encrypted numbers and we will get a encrypted result back which can only be decrypted by the user. Though multiplication or division of two encrypted numbers, and doing exponent calculations are still not possible. To overcome these limitations a lot of work is being put in developing Fully homomorphic encryption schemes [3]. But For this we are using an excellent python library from Paillier daylight society [4] that supports PHE with positive and negative integers, floats and strings.

We incorporated four algorithms from four different domains of machine learning. They needed to be modified so that they can work with encrypted data. Now InfoCrypt supports Linear Regression [5], Logistic regression [6] [7], k means clustering [8] and Artificial neural networks [9].

II. Limitations of traditional approach

The traditional approach of using machine learning involves using data in the form of simple plaintext for computational purposes. This poses many security concerns:

- Any company that opts to use cloud storage could forfeit their data to hackers or business opponents, which would become a significant loss to the company's future prospects.
- Technology has advanced to the point where the discussion of laws regarding privacy, safety and boundaries of individuals is now a critical concern with regards to the internet.
- Social networking sites, e-commerce sites and search engines keep track of all user interactions and information on their sites. While the companies who manage these sites claim that this is to improve user experience by providing better recommendations, this brings up a pressing concern regarding the extent that this information can be used.
- Current development of Microsoft SEAL can only perform bit-level computation, which is not sufficient for performing computations on user data.

III. Objective

Our primary objective is to protect data from unauthorized third party companies. Hence, we propose to encrypt the data being sent to the services. If encryption achieves data confidentiality, then it would also limit the possible reuse or processing of outsourced data as well as the sharing of data. Thus, the data owner will regain control over his/her confidential information. In this case, the services that involve data processing will also have to adapt to accommodate encrypted data.

IV. System design

A. Architectural design

- This is the main abstraction of the project. Each block indicates an independent process that takes input from the previous process. The User interacts with only the User block. This block is present in the user system and acts as the intermediate between the user and 3rd party applications. The user block processes input, encryption and decryption and as well as formulate requests.
- The third party application takes the input from the user block and sends it to their server. The target server processes all the data with the required machine learning model and generates the result. The result is then sent back to the user block which then decrypts the response and shows the result to the user.

Fig. 1. System Architecture layout

B. Components design

1) User block

The user block consists of the input form which includes model selection and input for the given model. It follows with an encryption and decryption block that encrypts and decrypts input or response data respectively. The encrypted input is in bytes and is sent to a request preprocessing block which converts and formulates the data from bytes to JSON and sends it to the network.

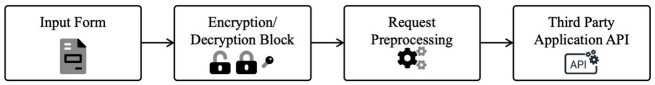

Fig. 2. User block component layout

2) Third Party Application

Any third party application that is confirmed to process encrypted data can interact with the user block and collects user input from the user block and sends the data directly to the target server to formulate the result using the required machine learning algorithms.

Fig. 3. Third Party Application component layout

3) Target Server

The target server is any third party server that can process encrypted data to produce results. It works with multiple algorithms on encrypted input and produces encrypted results that are sent back to the client module.

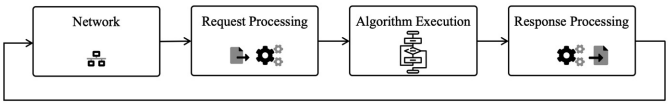

Fig. 4. Target Server component layout

V. Implementation

A. Client - Outgoing Request

The user interacts with the user block by selecting the model of analysis and by providing the required data as input. The data is then encrypted and is converted from bytes to JSON data by a request to the pre-processor. This is done to make it easier to send over the network. If any error occurs, the user is guided back to the input form and an error message is shown. If no errors have occurred, the data is sent to the 3rd party application via its API.

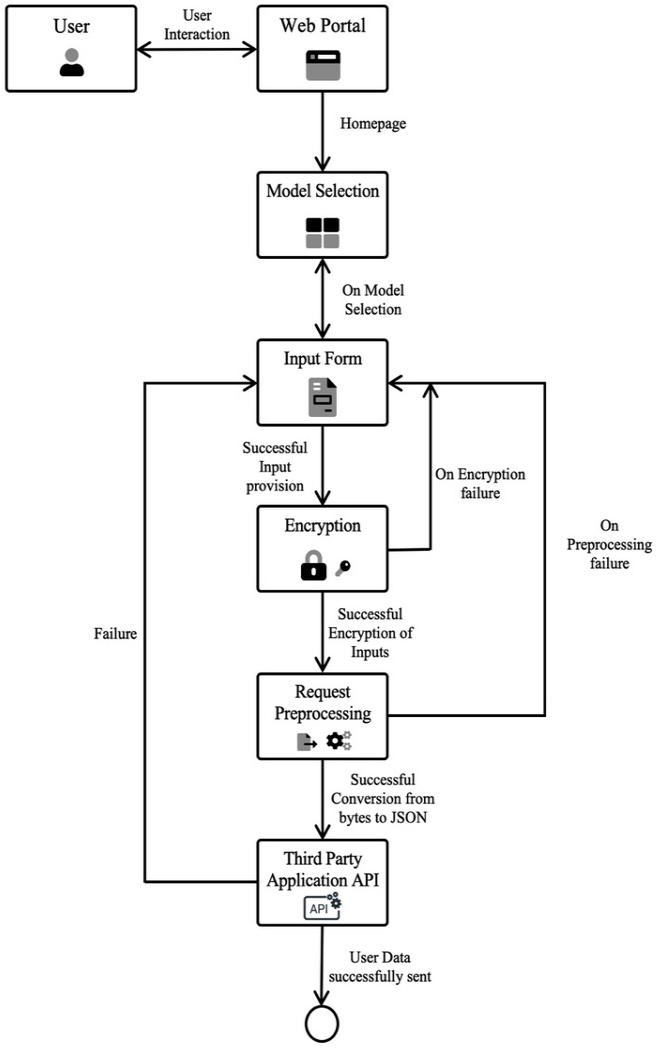

Fig. 5. Client Outgoing Request behavioural diagram

B. Target Server - Request Handling

The data sent over the network by 3rd party application arrives at the targeted server. The request received is then parsed back to byte data for processing from JSON in the request processing block. The byte input data is then used in the machine learning algorithms and the result is formulated which is encrypted. The encrypted result is then again converted into JSON data from bytes so that it can be transferred over the network to the user.

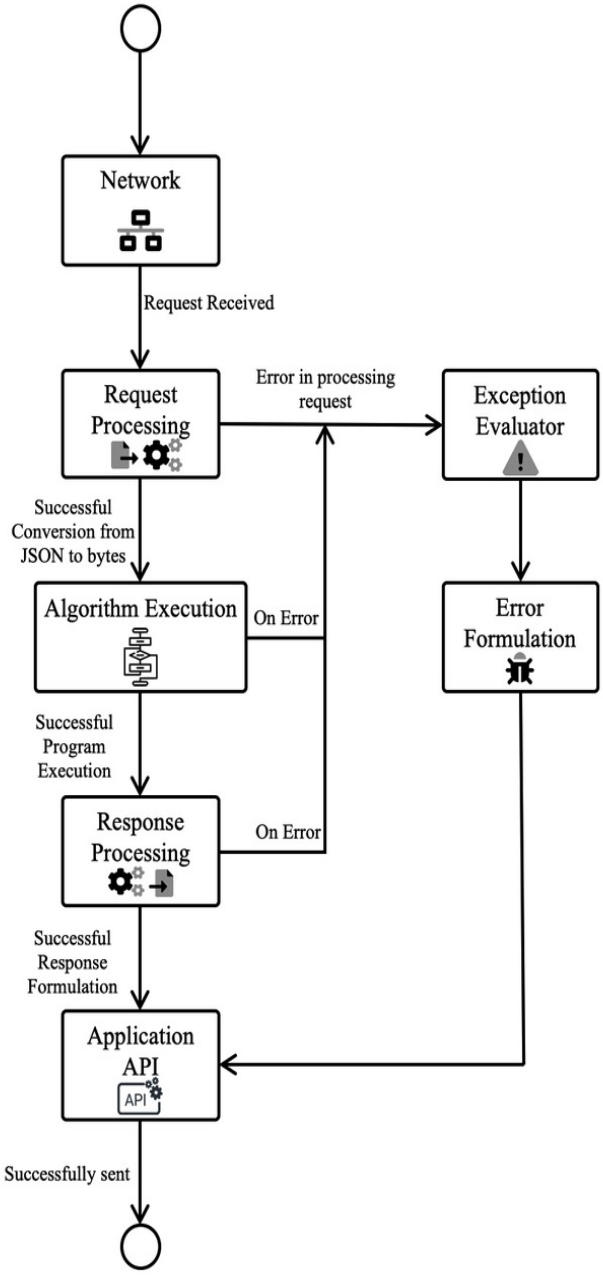

Fig. 6. Target Server Request Handling behavioural diagram

C. Client - Incoming Response

When the response is received by the user block it reformulates the data to bytes and decrypts it. If the response is in turn a request to process data for further execution of the algorithms on the server, the request is processed and encrypted and sent back to the server. If the response is the result to the query, the result is then shown to the user.

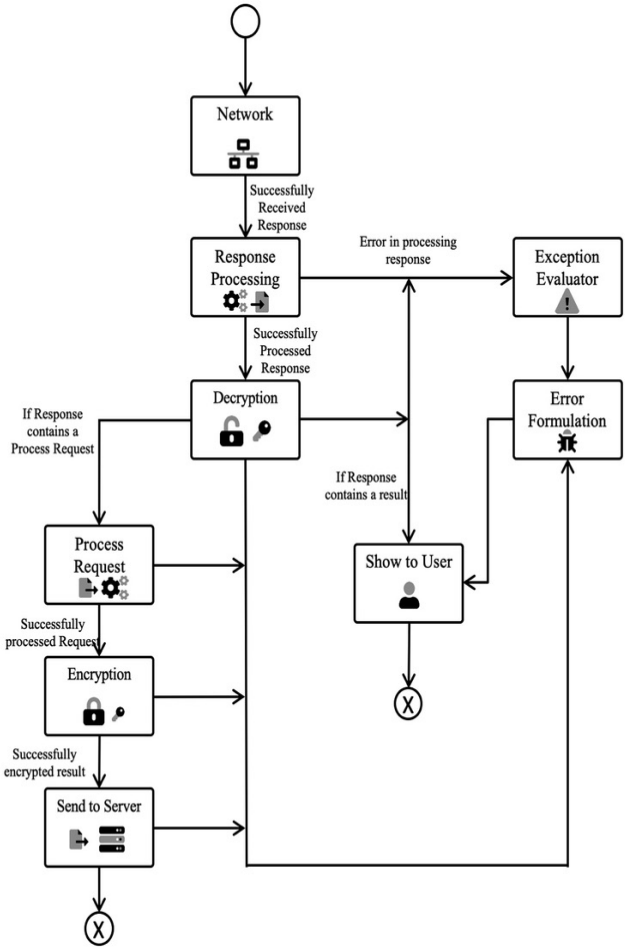

Fig. 7. Client Incoming Response behavioural diagram

VI. Results

As InfoCrypt is primarily a proof of concept, implementation of the working principle and the algorithm has been displayed in the form of five screens. The common component for these screens is the progress bar right below the logo.

A. Screen 1 - Input:

The left-hand side of the screen presents the user with four models to choose from Linear Regression, Classification, Clustering, and Artificial Neural Network. The respective information and form about each model are shown on the right-hand side, according to the user's choice.

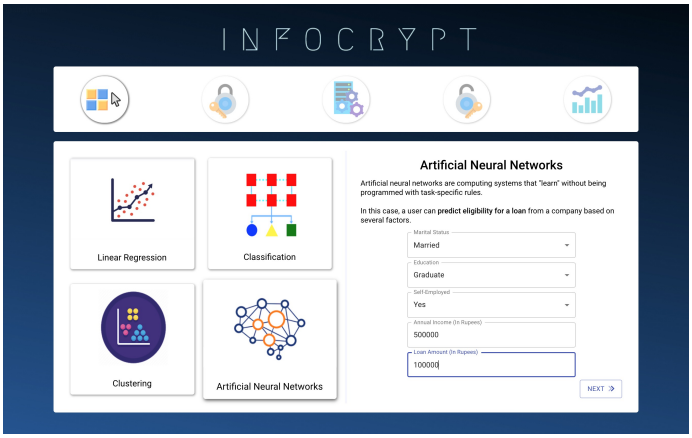

Fig. 8. Input Screen

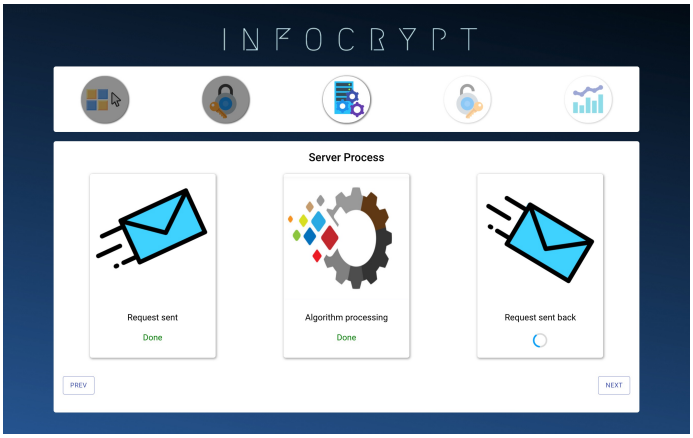

Fig. 10. Server Processing Screen

B. Screen 2 - Encryption:

On clicking 'Next' from the previous screen, the server creates two requests for comparison. One request is in encrypted form; the other request contains the data in plain text format. Apart from displaying both requests, the keys generated for the purpose of encryption/decryption are also displayed to the user.

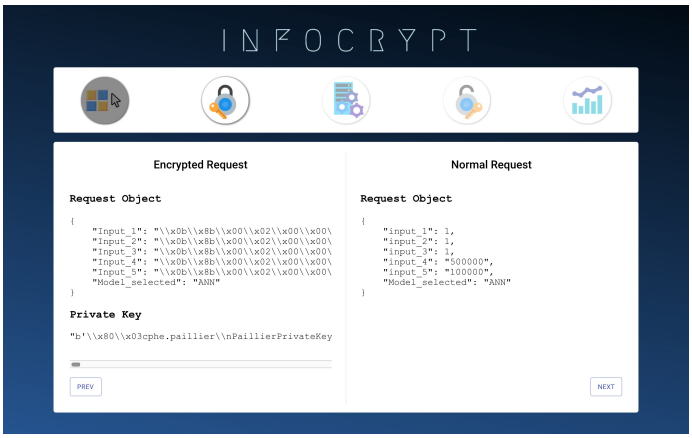

Fig. 9. Encryption Screen

C. Screen 3 - Server Process:

The third screen showcases the steps taken by the server to which the request is sent. As the process is rather complex, they have been abstracted into the main three parts - request received by the server, application of the algorithm upon the data sent, and the sending of the response by the server. As explained above, for the sake of comparison, two requests are sent to the server for processing.

D. Screen 4 - Decryption:

Once the application receives the response, the user can proceed to view the encrypted and decrypted response. In addition, the response received on handling the plaintext request is also available to view. As theorized, there is no difference between the decrypted output and the plaintext output. Hence, this confirms that there will not be any discrepancies in the output because of the additional encryption and decryption performed upon the data.

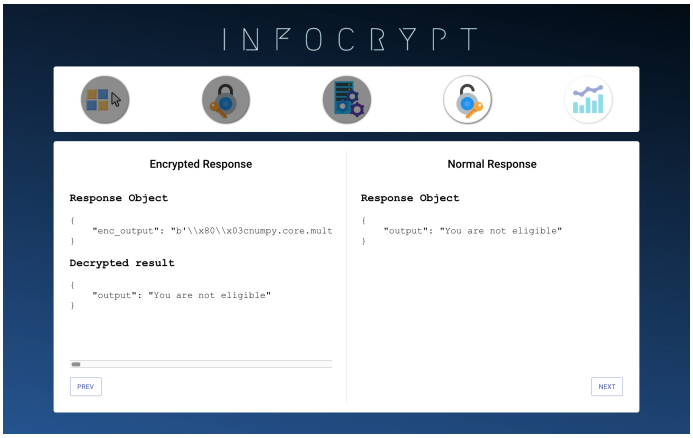

Fig. 11. Decryption Screen

E. Screen 5 - Performance Analysis:

The final screen presents the statistics to the user - the difference in five performance factors between the two approaches used (Traditional and InfoCrypt). Clearly, InfoCrypt requires extra time for the purpose of key generation, encryption, and decryption of data. Overall, it does take greater time for processing when compared to the traditional approach. The extra time taken for encryption and decryption is highlighted in red at the bottom of the screen. This is the tradeoff that the user will have to make every time this system is used - ensuring privacy in exchange for time. However the time taken for key generation is something that will only be required once unless the user opts to create a new public-private key set.

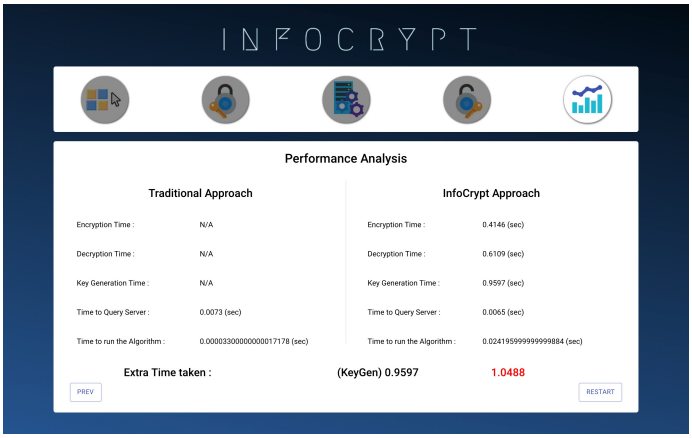

Fig. 12. Performance Metrics Screen

VII. Conclusion

This implementation lays the groundwork for designing ecosystems where the user data privacy can be in the hands of the users rather than trusting it with any third party companies. In this implementation we have shown that with little effort and changes on both user and server side algorithms, we can acquire the same results with much more privacy. Though the system is not robust and has many potential failures and also the extra time taken by Encryption and Decryption increases load and latency but these can be worked with Homomorphic encryption optimizations [10] [11] [12].

Acknowledgment

This research would not have been possible without the guidance, assistance and suggestions of many individuals. We would like to express our deep sense of gratitude and indebtedness to each and everyone who has helped us. We express our sincere gratitude to Dr. Mohan Babu G.N, Principal, BMSIT & M for providing all the facilities and the support. We heartily thank Dr. Anil G N, Head of Department, Department of Computer Science and Engineering, BMSIT & M for his constant encouragement and inspiration in taking up this topic for research. We gratefully thank our guide, Durga Bhavani A, Assistant Professor, Department of Computer Science and Engineering, BMSIT & M for encouragement and advice throughout the course of the research. Special thanks to all the staff members of the Computer Science Department and colleagues for their help and kind cooperation.

References

[1] P. Paillier, “Paillier Encryption and Signature Schemes,” Encyclopedia of Cryptography and Security, pp. 453–453.

[2] Paillers Algorithm Accessed on: May. 20, 2020. [Online]. Available: https://asecuritysite.com/encryption/pal_ex

[3] ”A look at Microsoft SEAL” Zhiniang Peng [2019] Accessed on: May. 20, 2020. [Online]. Available: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/323.pdf

[4] “ Python Paillier library ” Accessed on: May. 20, 2020. [Online]. Available: https://paillier.daylightingsociety.org/about

[5] ”Linear-Regression on Packed Encrypted Data in the Two-Server Model” Adi Akavia, Hayim Shaul [2018] Accessed on: May. 20, 2020. [Online]. Available: https://eprint.iacr.org/2019/1238.pdf

[6] ”Secure Logistic Regression Based on Homomorphic Encryption” Miran Kim,Yongsoo Song,Shuang Wang , [2018] Accessed on: May. 20, 2020. [Online]. Available: https://eprint.iacr.org/2018/074.pdf

[7] ”Doing Real Work with FHE: The Case of Logistic Regression” Jack L.H.Crawford, Craig Gentry, Shai Halevi [2018] Accessed on: May. 20, 2020. [Online]. Available: https://eprint.iacr.org/2018/202.pdf

[8] ”Sub-Linear Privacy-Preserving Near-Neighbor Search” M. Sadegh Riazi, Beidi Chen, Anshumali Shrivastava, [2019] Accessed on: May. 20, 2020. [Online]. Available: https://arxiv.org/pdf/1612.01835.pdf

[9] ”CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy” Nathan Dowlin, Ran Gilad-Bachrach, Kim Laine [2019] Accessed on: May. 20, 2020. [Online]. Available: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/04/CryptonetsTechReport.pdf

[10] ”Efficient Homomorphic Comparison Methods with Optimal Complexity” Jung Hee Cheon, Dongwoo Kim and Duhyeong Kim [2019] Accessed on: May. 20, 2020. [Online]. Available: https://eprint.iacr.org/2019/1234.pdf

[11] “Encryption Performance Improvements of the Paillier Cryptosystem ” Christine Jost, Ha Lam, Alexander Maximov, and Ben Smeets Accessed on: May. 20, 2020. [Online]. Available: https://eprint.iacr.org/ 2015/864.pdf

[12] “ Efficient paillier Crypto processor for privacy-preserving data mining ” Ismail San, Nuray At, Ibrahim Yakut, Huseyin Polat Accessed on: May. 20, 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1002/sec.1442