Homomorphic Encryption as End-to-End Solutionfor Smart Devices

Shanthala P T,
Research Scholar, Department of Computer Science & Engg.,
PESIT-Bangalore South campus, Visveswaraya Technological
University, Belagavi, Karnataka, India
shanthalapt@pes.edu

Dr. D Annapurna,
Professor and Head Department of CSE and ISE , PESIT-
Bangalore South campus, Bengaluru, Karnataka, India
annapurnad@pes.edu

Sravanthi Nittala, Arpitha S Bhat and Aishwarya
N.R. PESIT-Bangalore South campus Bengaluru,
Karnataka, India
Email: ndpsravanthi@gmail.com, arpithabht12@gmail.com, aishwaryaraman2000@gmail.com

Abstract
The recent past has seen a tremendous amount of advancement in the field of Internet of Things (IoT), allowing the influx of a variety of devices into the market. IoT devices are present in almost every aspect of our daily lives. While this increase in usage has many advantages, it also comes with many problems, including and not limited to, the problem of security. There is a need for better measures to be put in place to ensure that the users' data is protected. In particular, fitness trackers used by a vast number of people, transmit important data regarding the health and location of the user. This data is transmitted from the fitness device to the phone and from the phone onto a cloud server. The transmission from device to phone is done over Bluetooth and the latest version of Bluetooth Light Energy (BLE) is fairly advanced in terms of security, it is susceptible to attacks such as Man-in-the-Middle attack and Denial of Service attack. Additionally, the data must be stored in an encrypted form on the cloud server; however, this proves to be a problem when the data must be decrypted to use for running computations. In order to ensure protection of data, measures such as end-to-end encryption may be used. Homomorphic encryption is a class of encryption schemes that allow computations on encrypted data. This paper explores the application of homomorphic encryption for fitness trackers.

Keywords:
Internet of Things, Security, Homomorphic en- cryption, Denial of Service, Decryption, cloud server.

1. Introduction

Internet of things is a system of interrelated and connected objects which are capable of collecting and transferring data through wired and wireless networks without human interven- tion. It helps people live and work smarter. It is consideredas the next evolution of the internet wherein every entity will have the potential to connect to the internet. In the upcoming years with the advancement in technology it will provide an advanced level of services and nearly change the way thepeople live.

With the rapid developments in IoT technology, the numberof connected devices are also increased. Since the main functionality of IoT devices depends on the software and hardware along with its connectivity, securing them is the most important task that should be taken care of. Therefore, providing connectivity with high level of security becomes oneof the main objectives of IoT security.

Security in IoT refers to providing confidentiality, integrity, authenticity, privacy and security of users' sensitive data. It is gaining importance not just due to an increase in number of devices but also due to the influx of security issues arising in the different IoT layers. The challenges especially arise in the physical, network and application layer as further explored in this section.

Physical damage, hardware failure, and power limitationsare a number of the challenges faced within the physical layer.Some of the attacks that the network layer is susceptible to include DoS attacks, sniffing, gateway attacks, and unauthorized access. Malicious code attacks, vulnerabilities and bugswithin the software pose a threat to the application layer [16].An important aspect of IoT is its application in the field of healthcare, from heart rate monitors to blood sugar monitors and even fitness trackers used by the general population. The data collected and communicated by such devices is highly sensitive and personal to the individual user. In such cases, security holds more importance than before. The case of fitnesstrackers was considered for this particular project. In order to ensure secure transmission and storage of data, a methodfor end-to-end encryption using homomorphic encryption is proposed.

Manuscript received June 5, 2022
Manuscript revised June 20, 2022
https://doi.org/10.22937/IJCSNS.2022.22.6.10

2. Related work

Security in IoT devices has been implemented in a variety of methods, from encryption protocols [9], to authentication protocols, blockchain and by making use of Machine Learningand Deep Learning [2].

Hussain, Hussain, Hassan and Hossain, [2] reported intheir survey the various ML and DL based methods to enhance IoT security, including the usage of algorithms such as SVM, Naive Bayes, Decision Tree, Random Forest. They also emphasize the problem of consumption of more processing power and energy for implementing ML and DL algorithms. The implementations covered are for various problems that arise in IoT, such as for authentication and access control methods like Q-Learning can be employed as shown in Xiao, et.al.[3], ML based techniques also find use in attack detection and mitigation as shown in [4-6]. DL-based methods have applications in mitigating Denial of Service and Distributed Denial of Service attacks, [7].

Prasitsupparote, Amonrat, Yohei Watanabe, and Junji Shikata [18] proposed a methodology to implement homomor- phic encryption in health care systems. They demonstrated the usage of homomorphic encryption in constrained devices by implementing libraries on Raspberry Pi along with a PC actingas a cloud server.

Fitness trackers themselves, employ end-to-end encryption such as AES encryption protocol as shown by Classen, et al. [9].The authors worked with a FitBit to investigate the security measures employed by the manufacturers. For symmetrickey encryption, a 128-bit secret key which is manufacturer- generated is used. Further, it was found that most models that were manufactured before 2015 had optional encryption only. A proprietary technique was seen to be employed for theauthentication and pairing process, which cannot be completed without logging into the FitBit smartphone application. Local pairing, however, was not secured. The paper also served the purpose of highlighting the importance of end-to-end encryption in fitness trackers.

Homomorphic encryption is one such encryption protocol that can be used for IoT [10,11]. Song, Hu, Zhao, [11] intro- duce a modified version of Gentry's homomorphic encryption [17] that can be implemented for IoT security. The author's improved the efficiency of the bootstrapping by optimizing its parameters. They also contribute a method by which the ciphertext modulus can be generalized to be used in a larger variety of situations.

A large number of IoT devices rely on cloud services for storage of data, and ML computations for manipulation of data; Sun, et al. [10] describes the implementation of an improved Fully homomorphic Encryption protocol for multiplicative homomorphic computations, decreasing the noise and modulus by making use of modulus switching and re- linearization techniques. The authors were able to implement classification algorithms like Naive Bayes on FHE encrypted data. Additionally, the processing was further optimized by SIMD used along with their FHE scheme.

Homomorphic encryption can be applied to different types of IoT applications by designing the homomorphic operations in an efficient manner for the encrypted data. Gahi, Guennoun, El-Khatib [12] proposed a prototype database that stored and queried data which was encrypted using fully homomorphic encryption scheme. The authors proposed the circuits requiredto carry out SQL queries such as SELECT and UPDATE on the encrypted data. While the processing time to carry outthe operations was found to be substantial, the work done by the authors shows the scope of homomorphic encryption for application in cloud based services.

Homomorphic encryption does not prove to be a hindrance in providing security from attacks such as code injection as shown in Sgaglione et al.[13], where a Signature-based Intrusion Detection System was implemented such that it processed encrypted data in order to prevent attacks. This ensures security while also maintaining the privacy of data.

Prasitsupparote, Amonrat, Yohei Watanabe, and Junji Shikata [18] proposed a methodology to implement homomor- phic encryption in health care systems. They demonstrated the usage of homomorphic encryption in constrained devices by implementing libraries on Raspberry Pi along with a PC acting as a cloud server.

Such applications of Homomorphic Encryption, along with existing methods of IoT Security, provide the required encouragement and guidance for us to implement Homomorphic Encryption as an end-to-end encryption scheme in fitness trackers.

3. Motivation of the work

The literature review conducted pointed towards the need for implementing methods to strengthen the security for IoT devices, especially at cloud level, and at the same time make it possible to carry out data processing. The proposed system aims to focus on this aspect without compromising on the functionality of the fitness device under consideration. The system aims to allow the usage of third party cloud services while maintaining privacy of the users and allowing the service providers to carry out analysis on the data flow.

4. Terminologies used

Homomorphic encryption schemes refer to those which allow evaluations to be carried out on encrypted data without first decrypting the data. This was first proposed by Rivest, et al. [1], and a fully homomorphic encryption scheme was introduced by Craig Gentry [17] and since, many different schemes have been implemented with varying degrees of efficiency and applicability.

Every scheme consists of different implementations in one or more of the following algorithms: Key Generation, Encryption, Decryption, or Evaluation algorithms. All four of these algorithms form the core implementation of any homomorphic scheme. Homomorphic Encryption schemes can be additiveor multiplicative or both, depending on which they can be classified as Fully Homomorphic, Somewhat Homomorphic, and Partially Homomorphic Encryption. These vary not only in terms of operations performed but also the number of operations that can be accurately performed.

Consider a homomorphic function E() that is used to encrypt messages, m1 and m2. Once the data has been encrypted, addition and multiplication computations can be carried out on the now encrypted data, E(m1) and E(m2). The decrypted result will be equal to the result as when the computation is carried out on the unencrypted data.

For two plaintext data, m1 and m2,

$$E(m1) + E(m2) =$$

$$E(m1 + m2)E(m1) *$$

$$E(m2) = E(m1 * m2)$$

In the above equations, E() represents homomorphic encryption function [1]. As shown, this has numerous applica-tions due to its capability of preserving privacy while allowingcomputations of data. Depending on the implementation and type of homomorphic encryption used, some noise maybeintroduced in the result, which may increase with the number of computations depending on the scheme used.

5. Proposed method

Taking into consideration the work done for end-to-end encryption in the field of internet of things, including the use of homomorphic encryption, we propose a model for providingincreased security for fitness trackers. There are three entities considered for the data transaction: the tracker device, theuser's smartphone and the cloud server. The communication channels exist between the tracker and the Smartphone, and between the Smartphone and the cloud server. The trackerand the smartphone are connected by a Bluetooth low energy standard 5.2 connection. The data transmission between the phone and the cloud takes place over a standard internet connection.

The cloud server is considered to be a semi trusted third party cloud server. This entails that the data must be securely transmitted and additionally, must be encrypted even when stored on the cloud server. As explained earlier, the usage of homomorphic encryption ensures decryption is not necessary in order to carry out computation on encrypted data. Thus, homomorphic encryption is employed in the system.

Specifically, the CKKS (Cheon Kim Kim Song [14]) schemeis used. This scheme is a partially homomorphic encryption scheme that can encrypt real as well as complex numbers. It al-lows computations on complex numbers in the form of vectors. CKKS scheme gives approximate results after decryption and computation. For example, the number zero after encryption followed by decryption may result in a fractional number of the order of 10^{-8}. Although the results will be approximate, theapplicability of this scheme for real numbers comes in use for the fitness tracker data, which usually consists of real numberssuch as distance, GPS coordinates, time, calorie count, heart rate, etc. The homomorphic encryption process itself consists of four major steps: key generation, encryption, decryption and computation.

In our proposed system, the key generation takes place on the smartphone. Once the public and secret keys have been generated, the public key is transmitted to the paired device, while the secret key remains with the smartphone. In the device, the various sensors collect and accumulate the data during any activity. This data is encrypted according to the CKKS scheme using the public key that has been transmitted to it. The encrypted data is then transmitted via Bluetoothto the phone, where some of the data, such as the heart rate, which need no additional computation, are decrypted and madeavailable to the user. The encrypted data is then transmittedto the cloud server over an internet connection.

Usually, the fitness tracker will have standard authenticationimplemented, like login using pin/password or any authentication method adapted by the application installed in thephone to authenticate itself to the cloud. The duration for which it will last depends on the type of authentication used bythe developers. For example, some authentication is restricted to a session and for each session before the data transferauthentication happens, while other method may involve usingnonce value or some key that can indicate the identity of the source and is included in each data packet transferred from the phone to cloud server. Most commonly used techniquesare MD5, digital signature and use of session key. So, in our proposed method for authentication, we are relying on thestandard login process.

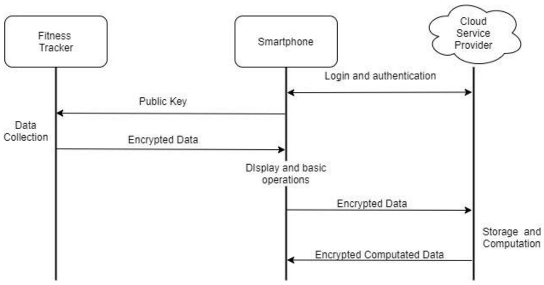


Fig. 1. Basic proposed System Architecture

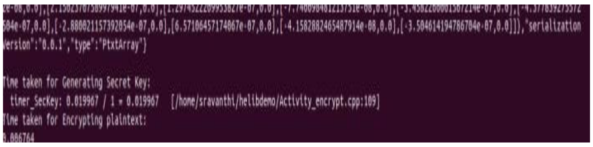
The cloud server carries out computations such as calculation of distance, speed, etc. on the encrypted data, without having to decrypt it. Since, CKKS supports only addition and multiplication, these operations are carried out homomorphically. Operations such as division are left to be computed by the smartphone device after decryption. Thus, if the total distance is to be obtained from the coordinates, the sum of squares, to be calculated according to the Euclidean distance formula, is computed at the server, but the square root of the obtained value is calculated on the decrypted plain text some on the smartphone.

As discussed in Hong,et al. [15], with respect to computa- tions on computer with 32 GB RAM, Intel Core i7-8700, CPU 3.2 GHz, the following results were computed. The authors reported that the bootstrapping gates took about 10.8 ms to evaluate except NOT and MUX gates. However, the executiontime increases with increase in input data bits. Since it involves many operations with respect to data and increases linearly with respect to data length and interactions.

The system specifications considered typically includes an ARM Cortex M3/M4 processor for the fitness tracker, based on the model of a typical fitness tracker. In this particular paper, an Intel Core-i5 processor is considered for carrying out the computation. The implementation involved the usage of the HELib library, and a comparison with the implementation using Microsoft SEAL for homomorphic encryption. The comparison is done in terms of time taken for tasks such as encryption, decryption, and secret key generation. Additionally,the Euclidean distance is also calculated using the HELib implementation.

6. Experimental result

We have obtained the results by running the HELib library for encryption on a dataset obtained from a fitness tracker. Thecomputation was carried out on a system with 8GB RAM, Intel Core-i5 and CPU 2.30 GHz. The data corresponds to the minutes where the user was fairly active over a specified period of time. While, a majority of the dataset is zero, the required part is a set of values that range between 2-48. The number of data items used are 1095, and the parameters used resultin a security level of 157.866. In order to demonstrate the capabilities of homomorphic encryption, once encrypted, this data is multiplied by a value of 1.5 and then decrypted to showthe result. The generation of Secret Key took 0.0199 seconds and encryption of dataset took 0.0676 seconds. The usage of CKKS schemes shows promising results for the application of such a system in fitness trackers.



```
2e-08,0.0],[2.1502378758997941e-07,0.0],[1.2974522209953827e-07,0.0],[-7.7468904812137 51e-08,0.0],[-3.4502266001367214e-07,0.0],[-4.377639275572
504e-07,0.0],[-2.880021157392054e-07,0.0],[6.57106457174067e-07,0.0],[-4.1582882465487914e-08,0.0],[-3.504614194786704e-07,0.0]]),"serialization
Version":"0.0.1","type":"PtxtArray"}

Time taken for Generating Secret Key:
  timer_SecKey: 0.019967 / 1 = 0.019967   [/home/sravanthi/helibdemo/Activity_encrypt.cpp:109]
Time taken for Encrypting plaintext:
0.086764
```

Fig. 2. Time taken for Secret Key Generation and Encryption

The same dataset and system specifications are considered for the implementation of the operations using the Microsoft SEAL toolkit. A Python based wrapper that implements the BFV scheme of homomorphic encryption was used for codingthe operations. The results are as shown in Table I, whichis comparison of the time taken for encryption, secret key generation and decryption on the dataset. This comparison wasdone to further highlight the performance of HELib against other available libraries for homomorphic encryption. Both SEAL and HELib show promising results in terms of speed of encryption, and decryption. HELib was then used for further experimentation as detailed below.

An additional operation considered is that of carrying out the calculation of Euclidean distance on two points in the x-y plane. This operation was carried out to check the feasibility of using homomorphic encryption for fitness trackers, where the Euclidean distance is used to calculate distance between GPS coordinates. The Euclidean distance calculation is divided into the following steps: first the subtraction is done before en-cryption, then the squaring and addition operations are carried out on the encrypted data, and next, the data is decryptedand the square root value is found. This is the proposed method for calculation of distance in fitness trackers where thecalculation occurs at the cloud level and the operations on the decrypted data occurs at the smartphone level. The distance values must be initially converted from GPS coordinates to UTM coordinates. The results are shown in Table II, and the time taken to complete this operation was 0.063522 seconds on the considered system.

TABLE I
Time taken (in seconds) for different actions carried out on adataset having 1095 data items.

Action	HELib	Microsoft SEAL
Secret Key Generation	0.0199 sec	0.0650 sec
Encryption	0.0676 sec	0.0608 sec
Decryption	0.050229 sec	0.0094 sec

Although the experiments carried out in this paper are with respect to a fitness tracker, the results remain same for any other smart devices that uses cloud based operations. The paper proposes an efficient way of encryption technique that can be used in smart devices aiming at a higher level ofsecurity for a user's data. Hence, the privacy of the data is taken care and is not compromised.

TABLE II
Results obtained for euclidean distance between two points, impleneted in HELib

Coordinates considered: (E:600.0 ,N:6100.0) , (E:400.0,N:6250.0)		
Operation	Expected value	Generated value
Squaring x coordinates	40000.00	40000.00
Squaring y coordinates	22499.99	22500.00
Addition	62499.99	62500.00
Final value (after decryp-tion)	249.99	250.00

7. Conclusion

The increasing interest in fitness has caused a massive increase in the usage of fitness trackers amongst people of all ages. This has led to a host of problems in terms of security and privacy. We have worked on this issue and proposed a method to alleviate the problem of privacy when data is stored on cloud servers. Homomorphic encryption is an encryption technique that caters to such situations as it can not just provide cryptographic protection but also, allow computationson encrypted data without having to decrypt the data first.

We believe that implementing such an end-to-end solution would enhance the security of the fitness tracker devices greatly. The HELib library shows immense potential to be applied for this purpose with its ability to generate near accurate values after computations. Further work can be done in providing authentication, integrity and confidentiality features over the encryption layer to further strengthen the data transmission channels.

References

[1] Rivest, R., Adleman, L., Dertouzos, M. "On data banks and privacy ho- momorphisms." Foundations of Secure Computation, pp.169–177 (1978).

[2] F. Hussain, R. Hussain, S. A. Hassan and E. Hossain, "Machine Learning in IoT Security: Current Solutions and Future Challenges," in IEEE Communications Surveys & Tutorials, vol. 22, no. 3, pp. 1686-1721, thirdquarter 2020, doi: 10.1109/COMST.2020.2986444.

[3] L. Xiao, Y. Li, G. Han, G. Liu, and W. Zhuang, "Phy-layer spoofing detection with reinforcement learning in wireless networks," IEEE Transactions on Vehicular Technology, vol. 65, pp. 10037–10047, Dec 2016

[4] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for internet of things," Future Generation Computer Systems, vol. 82, pp. 761 – 768, 2018.

[5] A. Abeshu and N. Chilamkurti, "Deep learning: The frontier for distributed attack detection in fog-to-things computing," IEEE Communications Magazine, vol. 56, pp. 169– 75, Feb 2018.

[6] S. Rathore and J. H. Park, "Semi-supervised learning based distributed attack detection framework for iot," Applied Soft Computing, vol. 72, pp. 79 – 89, 2018.

[7] N. Vlajic and D. Zhou, "Iot as a land of opportunity for ddos hackers," Computer, vol. 51, pp. 26–34, July 2018.

[8] S. Alharbi, P. Rodriguez, R. Maharaja, P. Iyer, N. Subaschandrabose, and Z. Ye, "Secure the internet of things with challenge response authentication in fog computing," in 2017 IEEE 36th International Performance Computing and Communications Conference (IPCCC), pp. 1–2, Dec 2017

[9] Jiska Classen, Daniel Wegemer, Paul Patras, Tom Spink, and Matthias Hollick. 2018. "Anatomy of a Vulnerable Fitness Tracking System: Dissecting the Fitbit Cloud, App, and Firmware". Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 2, 1, Article 5 (March 2018), 24 pages. DOI: https://doi.org/10.1145/3191737

[10] X. Sun, P. Zhang, J. K. Liu, J. Yu and W. Xie, "Private Machine Learning Classification Based on Fully Homomorphic Encryption," in IEEE Transactions on Emerging Topics in Computing, vol. 8, no. 2, pp. 352-364, 1 April-June 2020, doi: 10.1109/TETC.2018.2794611.

[11] Wei-Tao Song, Bin Hu, Xiu-Feng Zhao, “Privacy Protection of IoT Based on Fully Homomorphic Encryption”, Wireless Communications and Mobile Computing, vol. 2018, Article ID 5787930, 7 pages, 2018. https://doi.org/10.1155/2018/5787930.

[12] Gahi, Youssef, Mouhcine Guennoun, and Khalil El-Khatib. “A secure database system using homomorphic encryption schemes.” arXiv preprint arXiv:1512.03498 (2015).

[13] L. Sgaglione et al., “Privacy Preserving Intrusion Detection Via Homomorphic Encryption,” 2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), Napoli, Italy, 2019, pp. 321-326, doi: 10.1109/WETICE.2019.00073.

[14] Cheon J.H., Kim A., Kim M., Song Y. (2017) Homomorphic Encryption for Arithmetic of Approximate Numbers. In: Takagi T., Peyrin T. (eds) Advances in Cryptology – ASIACRYPT 2017. ASIACRYPT 2017. Lecture Notes in Computer Science, vol 10624. Springer, Cham. https://doi.org/10.1007/978-3-319-70694-8 15

[15] Hong, Mi Yeon, Joon Soo Yoo, and Ji Won Yoon. “Homomorphic Model Selection for Data Analysis in an Encrypted Domain.” Applied Sciences 10.18 (2020): 6174.

[16] Elrawy, M., Awad, A. and Hamed, H. Intrusion detection systems for IoT-based smart environments: a survey. J Cloud Comp 7, 21 (2018). https://doi.org/10.1186/s13677-018-0123-6

[17] Gentry, Craig. “Fully homomorphic encryption using ideal lattices.” Proceedings of the forty-first annual ACM symposium on Theory of computing. 2009.

[18] Prasitsupparote, Amonrat, Yohei Watanabe, and Junji Shikata. “Imple- mentation and analysis of fully homomorphic encryption in wearable devices.” The Fourth International Conference on Information Security and Digital Forensics. The Society of Digital Information and Wireless Communications. 2018.