Secure Tourist Tracking Using Fully Homomorphic Encryption in a Robotics Information System

Lujin Dersani and Ihab Elaff *

Posted Date: 30 June 2025

doi: 10.20944/preprints202506.2372.v1

Keywords: Fully Homomorphic Encryption; Secure Database; Robotic Information Systems

 doi:10.20944/preprints202506.2372.v1

Article

Secure Tourist Tracking Using Fully Homomorphic Encryption in a Robotics Information System

Lujin Dersani [1] and Ihab Elaff [2,*]

[1] Institute of Science / Cyber Security, Üsküdar Üniversitesi, Istanbul , Türkiye
[2] Computer Engineering Department, Üsküdar Üniversitesi,. Istanbul, Türkiye
* Correspondence: ihab.elaff@uskudar.edu.tr

Abstract

Tourists often face challenges such as language barriers and limited access to reliable information. To address this, a Robotics Information System (RIS) was developed using a humanoid robot named Arslan to assist visitors in real-time. The system manages sensitive data, including travel history, requiring strong encryption. Traditional methods like DES and AES were found inadequate, prompting the adoption of Fully Homomorphic Encryption (FHE), which enables computations on encrypted data without decryption. A mathematical tracking model encodes tourist movements securely. Performance tests show that while FHE introduces computational overhead, it ensures high privacy and resilience against modern attacks. This approach enhances tourist experience and ensures secure, real-time data handling in smart tourism environments.

Keywords: fully homomorphic encryption; secure database; robotic information systems

I. Introduction

These days, wealthy nations view the tourist industry as one of the key pillars supporting their economies and promoting their cultures. The most significant issues that tourists encounter, however, are the language barrier and the dearth of adequate information regarding each nation's tourist attractions. A Robotics Information System (Figure 1) was created in order to address these issues [1]. With the aid of vision [2], audition [3], speech, and other abilities, the humanoid robot Arslan [1] can engage with people from the moment of their arrival until their departure (they will be present at airports, hotels, hospitals, etc.).

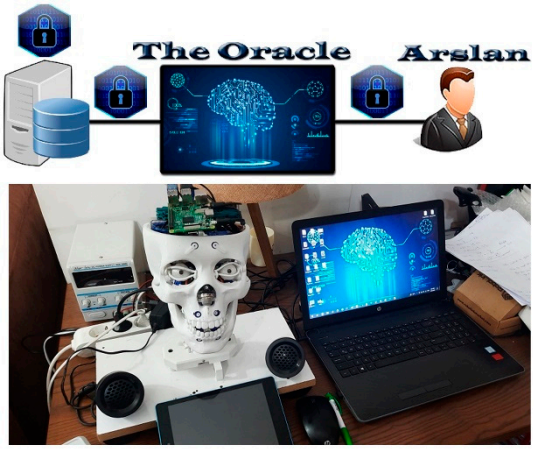

Figure 1. Robotics Information System (RIS) [1].

 doi:10.20944/preprints202506.2372.v1

Since the system's foundation is the provision and storage of tourist data in its database (such as personal data and travel schedules), the security of this sensitive data must be guaranteed by strengthening the RIS's database security.

Traditional encryption methods like the Data Encryption Standard (DES) and the Advanced Encryption Standard (AES) were assessed for their possible application in protecting communication between the database and the Oracle system during the early phases of system development. To enable safe data transfer between the two systems, a shared encryption key was created.

Nevertheless, a number of restrictions were found during the testing stage. Due to its vulnerability to compromise utilizing contemporary computational powers, "DES, being an older cryptographic algorithm with a relatively short key length (56 bits), was found to be particularly vulnerable to brute-force attacks" and is generally regarded as outdated [4]. Although AES provides more robust encryption with varying key lengths (128, 192, or 256 bits), questions have been raised about its long-term robustness [5]. Particularly, "if structural vulnerabilities within AES are discovered or if advancements in quantum computing materialize, the algorithm may be exposed to potential attacks" [6]. "Reliance on encryption methods that are susceptible to future compromise is deemed unsuitable" [7] considering the extremely sensitive nature of the material being handled. Consequently, it was determined that conventional symmetric encryption methods like DES and AES do not offer a strong enough degree of security to safeguard this type of sensitive data, especially in systems that need to be highly confidential and resistant to sophisticated attacks.

A public key and a private key are the two encryption keys used in asymmetric encryption algorithms, which are frequently employed for safe data transfer [8]. In this case, an oracle system encrypts data before transmitting it using the public key. The database receives the encrypted data and decrypts it using the matching private key [9]. Following decryption, the database sorts and encodes the data, among other necessary calculations, before storing it. This cryptographic technique has drawbacks even if it provides noticeably greater security than conventional symmetric encryption techniques. Though it still has some weaknesses, "asymmetric encryption is theoretically more resistant to brute force attacks due to the complexity of key generation and separation" [10]. The ultimate threat posed by "advances in quantum computing" [12] or "private key exposure, man-in-the-middle attacks" [11] are examples of these vulnerabilities. The development and security of user tracking mechanisms will be the main emphasis of this study.

II. Methods

It is crucial to start by describing and elucidating the many techniques and strategies that were used during the process in order to give a thorough grasp of the solutions created to meet the identified issue. These techniques were chosen and used with care, taking into account the unique needs and difficulties of the system in question.

A. Database Arrangement:

As can be seen in the tables below, the database first divides the places that tourists are anticipated to visit into tables. In the database structure, the tourist attractions that are anticipated to draw tourists are first categorized and organized in distinct tables. As demonstrated in Tables 1–3, this classification makes data administration and retrieval more effective.

Table 1. Example of List of Hospitals.

List of Hospitals				
Code No.	Hospital Name	Location	Contact Information	Working Hours
001	Hospital 1	LH1	05xxxxxx	xx:xx … xx:xx

 doi:10.20944/preprints202506.2372.v1

002	Hospital 2	LH2	05xxxxxx	xx:xx … xx:xx
003	Hospital 3	LH3	05xxxxxx	xx:xx … xx:xx

Table 2. Example of List of Restaurants.

List of Restaurants				
Code No.	Restaurant Name	Location	Contact Information	Working Hours
101	Restaurant 1	LR1	05xxxxxx	xx:xx … xx:xx
102	Restaurant 2	LR2	05xxxxxx	xx:xx … xx:xx
103	Restaurant 3	LR3	05xxxxxx	xx:xx … xx:xx

Table 3. Example of List of Museums.

List of Museums				
Code No.	Museum Name	Location	Contact Information	Working Hours
201	Museum 1	LM1	05xxxxxx	xx:xx … xx:xx
202	Museum 2	LM2	05xxxxxx	xx:xx … xx:xx
203	Museum 3	LM3	05xxxxxx	xx:xx … xx:xx

B. Mathematical Tracking System:

Following the database's organization of the tourist destinations, this input data is then systematically incorporated into the robotics information system for real-world application. In order for the visitor to make a choice, the robotics system must first issue a request to access particular database tables that hold pertinent geographic data. The system obtains the location code that corresponds to the tourist's first selection and multiplies it by one million. When a second site is chosen, its code is added to the original result after being multiplied by 1,000. The cumulative total from the first two phases is then immediately increased by the code of the third place that was chosen (as illustrated in Figure 2). The precise order of the tourist-selected destinations is reflected in a single number identity generated by this computational process. The system can more effectively and precisely track and react to the visitor's movements since each element of this number corresponds to a distinct location and is arranged chronologically based on the visitor's choices.

$$002 \rightarrow 101 \rightarrow 203$$

$$002*1000000 + 101 * 1000 + 203 = 002101203$$

 doi:10.20944/preprints202506.2372.v1

Figure 2. User Tracking Example.

It is essential that the data produced by the previously explained encoding method not be stored in its raw form due to the sensitivity of the resulting data. Before each storage procedure, encryption must be used to guarantee data security and shield it from unwanted access. As a result, two different encryption techniques have been put out to protect the data both during storage and transmission:

Proposal 1:

This method entails sharing an encryption key between the database and the Oracle system. The database uses this key to encrypt the requested data before transferring it when Oracle makes a data request. Oracle receives the encrypted data, decrypts it locally, adds or processes any information that is required, and then uses the same key to re-encrypt the updated data. Later, the database receives the re-encrypted data, decrypts it, and stores it safely. Both symmetric and asymmetric encryption algorithms will be applied in this situation.

Proposal 2:

The second strategy, on the other hand, gives Oracle sole responsibility for encryption. By limiting access to the encryption mechanism, Oracle improves data security in this situation by encrypting data using a private key that is not shared with the database. After that, the database receives the encrypted data and applies the required mathematical change without decrypting it. The outcome of processing and encryption is immediately saved in the database. This approach improves privacy and control by guaranteeing that private data is not accessible to the database itself. Fully homomorphic encryption, or homomorphic encryption, is used to achieve this.

Any possible breach might result in major privacy violations and unapproved exploitation due to the extremely sensitive nature of the data involved, especially that which is linked to tracking travelers' movements and trip histories. As a result, even with enhanced security, systems handling private information and real-time location data might not be adequately protected by asymmetric encryption.

C.Homomorphic Encryption Techniques:

Symmetric encryption, with a focus on Fully Homomorphic Encryption (FHE), was the last encryption technique investigated during system development. This method eliminates the need for decryption altogether by allowing computations to be done directly on encrypted data. According to earlier studies, "Homomorphic encryption systems can compute on encrypted data without requiring the private key." The result of these computations is encrypted in and of itself. Any calculations performed on the encrypted data yield the same outcome as when the data was raw. [13] Oracle used a private key that was only known to it to control encryption and decryption in this implementation. Confidentiality was maintained during the safe transmission and database storage of the encrypted data.

Without having access to the raw, unencrypted data, the database then carried out the required calculations, including structuring and categorizing the data. Sensitive information was protected throughout the data lifecycle by securely storing the processed and encrypted data in the database.

When compared to other encryption methods that have been examined, such symmetric (DES and AES) or asymmetric encryption, this approach showed notable gains in security and privacy. Enhanced resilience to a variety of assaults, such as those targeting key compromise or data interception during transmission, is provided by fully homomorphic encryption. Applications requiring sensitive and privacy-sensitive data, such monitoring visitor movements and managing private location data, are especially well-suited for its capacity to facilitate secure calculations on encrypted data.

 doi:10.20944/preprints202506.2372.v1

III. Performance & Results Analysis

This section provides a thorough analysis of the encryption procedure used in the project, along with particular instances that highlight its usefulness. Additionally, we go into the results of its application as well as the challenges faced throughout the implementation stage.

A. User Tracking Case:

In the first phase of the user-information system interaction scenario, the system asks the database for a list of tourist destinations that have been saved, enabling the traveler to choose a preferred place. Because the list is encrypted using Fully Homomorphic Encryption (FHE) techniques, which allow computations to be conducted directly on encrypted data without the need for decryption, this data exchange takes place inside a framework that guarantees user privacy and data security.

The database makes the required calculations after receiving the encrypted data in order to organize the locations from oldest to newest using an arithmetic sequence depending on the time each location was added. This computing structure allows for accurate processing in a fully encrypted environment while maintaining the data's complete confidentiality.

Several real-world tests were carried out to verify the veracity of the findings and evaluate the system's operation in diverse scenarios in order to illustrate the efficacy of this strategy. The effectiveness of the system and its appropriateness for safe tourist information systems are assessed in part by these tests. The graphs show how the key size, the number of steps, and the system's encryption time relate to one another. The association between the number of steps and the key size is displayed in the first graph (Figure 3). the relationship between the step size and the key size (in bits). Three main categories of key sizes are clearly visible: 1024 bits, 2048 bits, and 4096 bits, each representing a different level of cryptographic strength. Upon examining the bars in this chart, it becomes apparent that the key size does not change linearly with increasing step size. Instead, the values appear in distinct, repeated levels at certain step intervals. This repetition suggests that the key generation process follows an algorithm that selects key sizes based on predefined ranges or thresholds, possibly governed by security requirements or algorithmic constraints. That this chart, we can conclude that the relationship between step size and key size is indirect and segmented, implying the influence of additional factors beyond step size in determining the final key length.

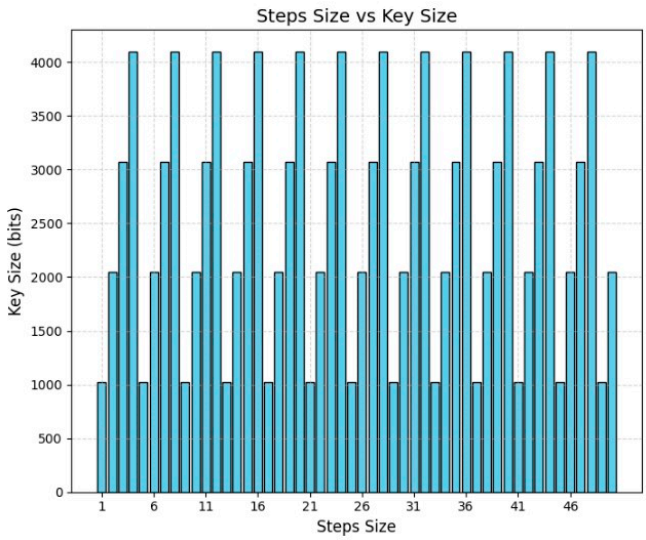

Figure 3. Steps Size Vs Key Size.

The second graph, presented in Figure 4, illustrates the relationship between execution time and step size. An approximately positive correlation was observed, indicating that as the step size

 doi:10.20944/preprints202506.2372.v1

increases, the execution time tends to increase as well. The bars exhibit noticeable variations in height, with some values exceeding three seconds while others remain below one second. This reflects significant differences in computational complexity, likely resulting from variations in the generated key size or the underlying mathematical operations. These fluctuations suggest that specific step sizes may lead to more computationally intensive processes. Therefore, while the relationship between step size and execution time is not perfectly linear, it is generally proportional—larger step sizes typically result in longer execution times due to increased computational demands.

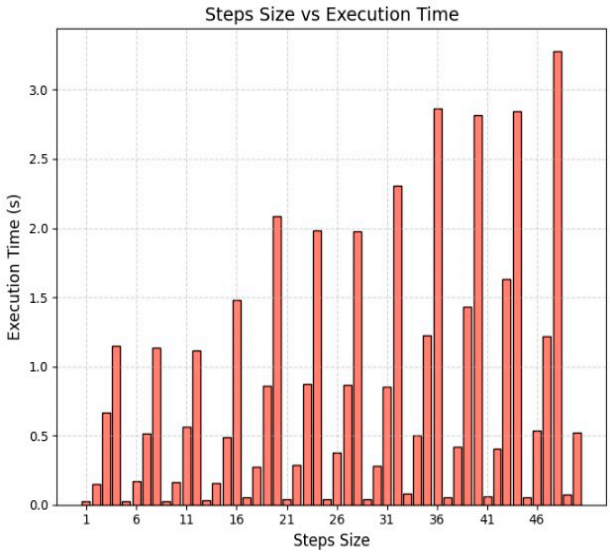

Figure 4. Steps Size Vs Execution Time.

B. Limitation:

During the system testing phase, several challenges were identified:

1. Non-Linear Scalability of Key Size with Step Size:

The plot (Steps Size vs. Key Size) reveals that key sizes do not increase linearly with step size. Instead, key sizes tend to cluster around fixed values—namely 1024, 2048, and 4096 bits—regardless of the specific step size. This irregular pattern suggests that the key generation mechanism operates based on predefined thresholds rather than gradual scaling. While the use of larger keys (2048 bits and above) enhances security, it also leads to a heavier ciphertext, resulting in increased communication overhead and storage demands. These effects become more pronounced in configurations that apply frequent encryption operations at high step sizes, limiting the system's scalability and efficiency.

2. Execution Time Spikes and Computational Overhead:

The plot (Steps Size vs. Execution Time) shows a general positive correlation between the increase in step size and execution time. However, the rise is not smooth—distinct spikes in execution time are observed at multiple step intervals, particularly near steps 6, 16, 21, 31, 36, and beyond. These sudden jumps in processing time indicate that certain step sizes trigger more computationally expensive operations, potentially due to underlying encryption complexity or increased input sizes. Such behavior introduces performance bottlenecks, especially for real-time or latency-sensitive applications. The inconsistency in execution time makes it difficult to guarantee predictable system performance.

These observations emphasize a fundamental trade-off between cryptographic strength and operational efficiency. While stronger encryption ensures better data security, it can significantly degrade performance if not managed through optimized key selection and step tuning. Therefore, to

 doi:10.20944/preprints202506.2372.v1

achieve an effective balance between security and responsiveness, systems must adopt adaptive encryption strategies that dynamically align key sizes and step sizes with performance requirements.

IV. Conclusion

By the end of this study, a thorough grasp of the several encryption methods that can be used to handle extremely sensitive data had been attained. Out of all the techniques examined, fully homomorphic encryption (FHE) was found to be the most appropriate strategy given the state of technology today. In addition to offering strong defense against common cyberthreats, such as man-in-the-middle assaults, this method makes it possible to do intricate calculations on encrypted data directly, doing away with the requirement for decryption altogether.

FHE's deployment was essential to protecting travelers' privacy, especially when it came to tracking their travels between locations. Additionally, by facilitating the safe processing of real-time data, this method improved the robotic information system's performance and decision-making capacities. In the end, it is anticipated that the project's results will enhance the overall traveler experience and benefit the tourism industry. By doing this, it encourages tourists to make happy, long-lasting experiences while also boosting economic development and raising awareness of local cultural assets around the world..

Given its suitability for protecting the kind of sensitive data involved in tracking tourist activities, we have chosen fully homomorphic encryption (FHE) as the main cryptographic technique at this point in the project. Our future development goals, however, involve looking into different encryption methods that provide more effective key management, especially ones that can accommodate a wider range of operations throughout several phases of travelers' trip monitoring. This approach seeks to preserve the highest standards of data integrity and secrecy while improving the system's scalability and overall performance.

References

1. I. Elaff, "Robotic Information System (RIS): Design of Humanoid Robot's Head Based on Human Biomechanics", El-Cezeri Journal of Science and Engineering, vol. 10, no. 2, pp. 420–432, 2023, doi: 10.31202/ecjse.1249294.
2. I. E. Guemmam and I. Elaff, "Human Face Localization in 3D For Humanoid Robot Vision," 2025 7th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (ICHORA), Ankara, Turkiye, 2025, pp. 1-4, doi: 10.1109/ICHORA65333.2025.11017068.
3. M. I. Al Karaki and I. Elaff, "Modelling Humanoid Robot Audition for Sound Source Localization Using Artifical Neural Network," 2025 7th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (ICHORA), Ankara, Turkiye, 2025, pp. 1-4, doi: 10.1109/ICHORA65333.2025.11017196.
4. B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C , 2nd ed. New York, NY, USA: Wiley, 1996.
5. J. Daemen and V. Rijmen, The Design of Rijndael: AES - The Advanced Encryption Standard . Berlin, Germany: Springer, 2002.
6. L. Chen et al., "Report on Post-Quantum Cryptography," National Institute of Standards and Technology (NIST), NISTIR 8105, 2016. [Online].Available:https://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8105.pdf
7. G.Alagic et al., "NIST Post-Quantum Cryptography Standardization," National Institute of Standards and Technology (NIST), 2022. [Online].Available:https://csrc.nist.gov/projects/post-quantum-cryptography
8. W. Stallings, Cryptography and Network Security: Principles and Practice , 7th ed. Upper Saddle River, NJ, USA: Pearson, 2016.

 doi:10.20944/preprints202506.2372.v1

9. A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, Handbook of Applied Cryptography . Boca Raton, FL, USA: CRC Press, 1996.
10. D. R. Stinson and M. B. Paterson, Cryptography: Theory and Practice , 4th ed. Boca Raton, FL, USA: CRC Press, 2018.
11. C. Mitchell, "Security implications of asymmetric cryptography in practice," IEEE Security & Privacy , vol. 15, no. 2, pp. 45–52, Mar./Apr. 2017.
12. L. Chen et al., "Report on Post-Quantum Cryptography," National Institute of Standards and Technology (NIST), NISTIR 8105, 2016. [Online].Available:https://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8105.pdf
13. Mahmood, Z. H., & Ibrahem, M. K. (2018). New fully homomorphic encryption scheme based on multistage partial homomorphic encryption applied in cloud computing. *2018 1st Annual International Conference on Information and Sciences (AiCIS)*, 182–186. https://doi.org/10.1109/aicis.2018.00043