CSEC 604 - Cryptography and Authentication
Implementation Project Report

ENHANCING DIGITAL TWIN SECURITY THROUGH PARTIAL HOMOMORPHIC ENCRYPTION

December 9, 2023

Vijayavarman Arunasalam Harikrishnan, Sree Alekhya Teerthala
Department of Computing Security
College of Computing and Information Sciences
Rochester Institute of Technology
va7457@rit.edu, st6626@rit.edu

1 Abstract

The advent of Digital Twins has enabled real-time data analysis on a massive scale. However, it has also introduced numerous security concerns. Data tampering, privacy leakage, privilege escalation, and visualization tampering are some common security issues that Digital Twins face. This project aims to address these challenges by implementing the ElGamal cryptosystem and utilizing its homomorphic property to perform computational operations on the data without having to decrypt it. By applying this cryptosystem to Digital Twins, we enable secure computations while preserving data privacy. We will demonstrate and address the security vulnerabilities in Digital Twins, using the healthcare domain as an illustrative example.

2 Introduction

Digital Twins are virtual clones of a physical object, system, or organization. They are connected to their corresponding physical entity in real time, mimicking everything from behavior to performance. Since the inception of the concept of Digital Twins in 2002 by Dr. Michael Grieves, a professor at the Florida Institute of Technology, it has gained significant traction across a wide range of sectors such as manufacturing, healthcare, transportation, smart cities, aerospace, energy and many more. The production and supply chain management of products can be digitized with the introduction of Digital Twins in the manufacturing domain, which reduces defects and errors. Enhanced resource allocation, grid resilience and modeling power grids are some benefits that Digital Twins bring to the table once implemented in the energy sector.[1] The best Digital Twin example in transportation is Ford Motor Company. They create their autonomous vehicles using Digital Twins. They evaluate the functionality of their self-driving algorithms and test their product in a variety of driving conditions. They can safely test their product in the virtual world without requiring a physical prototype. This helps reduce the price and duration of development.

A whole new world of potential in medicine is made possible by Digital Twins. Medical practitioners can customize medications and therapies for each patient. Pfizer, a major pharmaceutical company, takes an average of 12 years to create novel medications and revolutionary treatments that improve the lives of patients. However, in 2020, just nine months after they committed to create a COVID-19 vaccination, Pfizer came up with a vaccine which the World Health Organization approved. This was made possible due to the incorporation of Digital Twins. Pfizer started building Digital Twins of real sites as a 3D production line scanning and significantly reduced the time to produce the vaccine.

Digital Twins are complex systems composed of four main layers: Layer 1 (data acquisition and transmission), Layer 2 (data synchronization), Layer 3 (data modeling), and Layer 4 (data simulation and representation). Layer 1 operates in the physical space, where sensors and real-world entities are located, while Layers 2 through 4 are built and tracked in the digital domain.[2]

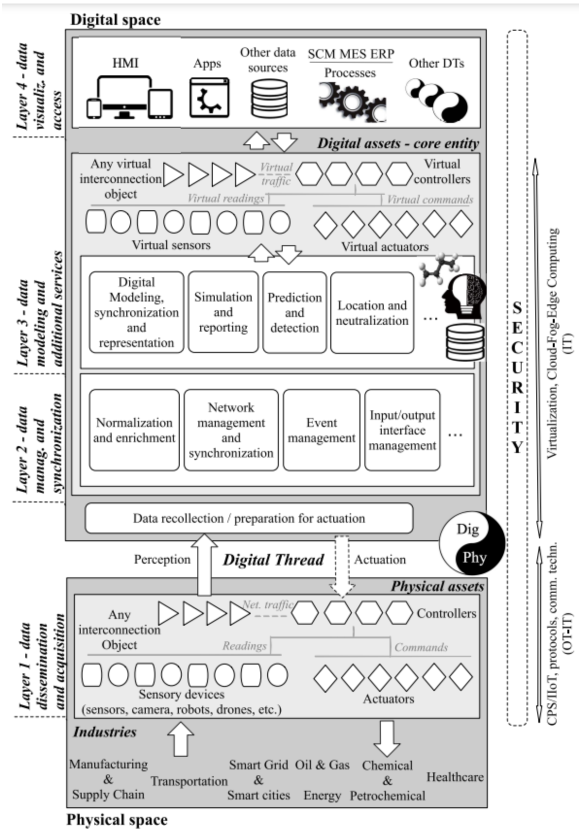

Digital space
Layer 4 - data visualiz. and access
HMI
Apps
Other data sources
SCM MES ERP
Processes
Other DTs
Digital assets - core entity
Any virtual interconnection object
Virtual traffic
Virtual controllers
Virtual readings
Virtual commands
Virtual sensors
Virtual actuators
Layer 3 - data modeling and additional services
Digital Modeling, synchronization and representation
Simulation and reporting
Prediction and detection
Location and neutralization
...
Layer 2 - data manag. and synchronization
Normalization and enrichment
Network management and synchronization
Event management
Input/output interface management
...
Data recollection / preparation for actuation
Dig
Phy
Perception
Digital Thread
Actuation
Physical assets
Layer 1 - data dissemination and acquisition
Any interconnection Object
Net. traffic
Controllers
Readings
Commands
Sensory devices (sensors, camera, robots, drones, etc.)
Actuators
Industries
Manufacturing & Supply Chain
Transportation
Smart Grid & Smart cities
Oil & Gas Energy
Chemical & Petrochemical
Healthcare
Physical space
SECURITY
Virtualization, Cloud-Fog-Edge Computing (IT)
CPS/IIoT, protocols, comm. techn. (OT-IT)

Figure 1: Architectural Design of a Digital Twin [2]

Layer 1 captures real-world dynamics and generates control commands for physical assets. In Layer 2, data is collected from sensors, processed, synchronized, and prepared for Layer 3 services. Layer 3 involves creating digital models using the preprocessed data, integrating necessary services, and implementing security features. Layer 4 enables end users to visualize the Digital Twin to draw conclusions and interpretations.

The security issues related to Layer 2 and Layer 3 are very concerning and have been discussed in the report as we move forward. To address the security concerns, we proposed utilizing the homomorphic property of Elgamal cryptosystem to perform calculations on the data in Layer 2 and Layer 3.

3 Literature Review

3.1 Security Issues in Digital Twins

Critical security issues for Digital Twins include the authenticity, integrity, and confidentiality of Digital Twin data, data privacy, and potential data tampering. As the adoption of Digital Twins increases across various fields, the security of these systems has become a top priority. Data is first transmitted from the physical entity (Layer 1) and then fed into the Digital Twin. This data transfer occurs securely using cryptographic algorithms. However, decryption is necessary in Layer 2 to perform computations, modeling, and visualizations in Layers 3 and 4. This leaves critical parameters and raw data generated by the Digital Twin unencrypted and vulnerable. The vulnerability of the data in Layers 2 and 3 has been mentioned very neatly in a paper that came up in 2022 which includes Man-in-the-Middle attacks, data tampering, extraction of sensitive information and potential privacy breaches. These vulnerabilities are especially concerning due to the critical nature of Digital Twins, which handle sensitive data and control essential operations.[2] Another challenge in this context is related to restricted data access. Conventional data encryption methods often require sharing decryption keys with all authorized users, even if they only need access to a small portion of the data. This sharing of keys can introduce security risks.[3]

3.2 Drawbacks of traditional security solutions for Digital Twins

Traditional approaches to address vulnerabilities in Layers 2 and 3 of Digital Twins include hashes, blockchains, IDS, firewalls, access controls, and cryptographic solutions. However, each of these solutions has limitations.[4]

- Access Controls - Access controls act as the gatekeepers of the data in Layer 2 and Layer 3, deciding who gets access to it and who doesn't. However, there is a huge potential for clever individuals to find loopholes or vulnerabilities in the access control mechanisms, bypassing the Firewall or IDS and performing Privilege Escalation.

- Cryptographic Solutions - Cryptographic solutions rely heavily on keys for encryption and decryption. A major challenge lies in the improper management or unauthorized access to these keys, which can compromise the integrity and security of the data protected by cryptographic protocols in Layers 2 and 3.

- Blockchain - Blockchain technology offers immutability, but this feature can be both an advantage and a disadvantage. It is difficult to change incorrect data that was entered due to human negligence in Layers 2 and 3 because of this immutability.

- Hash Functions - Hash functions offer a unique identifier for each piece of data. However, collisions can occur, where two different pieces of data generate the same hash value. This undermines the reliability of the hash function, as it should ideally generate a unique identifier for each piece of data.

3.3 Partial Homomorphic Encryption using ElGamal Algorithm

Homomorphic encryption is one solution that can be considered to address the above-mentioned security issues with Digital Twins. Computation and analysis of data generated by Digital Twins can be done without decryption, preserving the confidentiality, authenticity, integrity, and privacy of the data. [5] This project aims to implement homomorphic mathematical operations on encrypted data, employing Partial homomorphic encryption schemes such as RSA, Paillier, and ElGamal. Among these options, Elgamal algorithm was chosen for homomorphic multiplication after carefully considering factors such as key generation time, encryption time, decryption time, memory usage, and encryption throughput performance metrics. Both the Paillier and ElGamal cryptosystems are generally preferred over unpadded RSA due to their inherent semantic security properties. However, Paillier's algorithm only supports homomorphic addition, but the multiplicative homomorphic property is the main focus of this project. Therefore, the Elgamal algorithm was chosen for this project.[6]

Key size= 1024 bit	**RSA**	**Paillier**	**Elgamal**
Key Generation Time (in nanoseconds)	2438544400	2719886000	1046796800
Encryption Time	62495700	453073000	62497400
Decryption Time	124991100	421844900	31245900
Memory Usage (in bytes)	273	530	452

Table 1: Comparative analysis of Homomorphic Algorithms [6]

4 Project idea

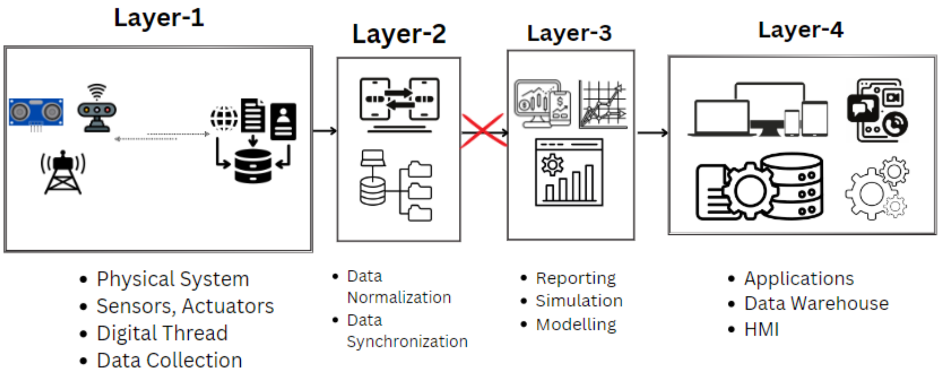


Figure 2: Existing issue in Digital Twins

The architecture of Digital Twin proves that the data in Layer 2 and Layer 3 presents security challenges and is defenseless. Some previously proposed solutions have also turned out to be defenseless. Existing vulnerabilities necessitate a solution, leading us to propose Elgamal homomorphic encryption. This encryption scheme allows for computations on encrypted data, shielding Layer 2 and Layer 3 from unauthorized access and modification. By implementing Elgamal homomorphic encryption, we aim to significantly improve the system's security and mitigate cyberattacks.

4.1 Proposed Solution

We consider the healthcare industry and look to implement our solution to solve security issues in Digital Twins. This involves creating a Digital Twin of the human heart and calculating the Cardiac Output through homomorphic encryption. This Digital Twin goes beyond just replicating the heart's subtleties, delivering insights into vital signs like heart rate and stroke volume. Now, to safeguard the privacy of this sensitive health data, we've introduced a sophisticated layer of security using ElGamal cryptography. This cryptographic method not only keeps the heart rate and stroke volume information confidential but also allows us to securely perform calculations on the data without revealing its actual values. By incorporating the unique properties of ElGamal, especially its multiplicative homomorphism, we can perform encrypted computations. In simpler terms, we can perform the multiplication operation on the encrypted heart rate and stroke volume data to calculate a key metric known as cardiac output. This innovative fusion of Digital Twin technology and advanced cryptographic techniques ensures not only the accuracy of cardiovascular data but also its highest level of privacy.

4.2 Elgamal homomorphism

To make sure that the patient's data is private once it is generated from the Digital Twin, we are performing ElGamal encryption on it.

In a cyclic group G of order q, with generator g, if the public key is (G,q,g,h), where $\mathrm{h}=\mathrm{g}^{\mathrm{x}}$ and x is the secret key, then the encryption of a message m is $\mathrm{E(m)}=(\mathrm{g}^{\mathrm{r}},\mathrm{m.h}^{\mathrm{r}})$ for some random r €{0,......,q-1}. The homomorphic property is [6]

$$\begin{aligned}\mathrm{E(m_1).E(m_2)} &= (\mathrm{g}^{\mathrm{r1}},\ \mathrm{m_1.h}^{\mathrm{r1}})(\mathrm{g}^{\mathrm{r2}},\ \mathrm{m_2.h}^{\mathrm{r2}}) \\ &= (\mathrm{g}^{\mathrm{r1+r2}},(\ \mathrm{m_1.\ m_2})\ \mathrm{h}^{\mathrm{r1+r2}}) \\ &= \mathrm{E(m_1.\ m_2)}\end{aligned}$$

5 Project Implementation

To bring our proposed solution and system to life, we'll be using two essential tools: Matlab and Jupyter Notebook. Firstly, Matlab is our platform for crafting a lifelike simulation of the human heart. It acts as our virtual canvas, allowing us to design and generate a Digital

Twin Simulator that accurately reflects the real dynamics of the heart and produces Heart Rate and Stroke Volume data.

Now, with Jupyter Notebook, we address the security and privacy concerns. We implement homomorphic encryption based on the ElGamal algorithm to ensure the confidentiality of the data generated by our Digital Twin. This encryption ensures that the confidentiality of the Heart Rate and Stroke Volume data is maintained while we seamlessly perform homomorphic encryption to generate the Cardiac Output.

In this system, Matlab orchestrates the heart's virtual performance, and Jupyter Notebook encrypts the data, allowing us to perform secure calculations without ever exposing sensitive details.

5.1 Project design and implementation

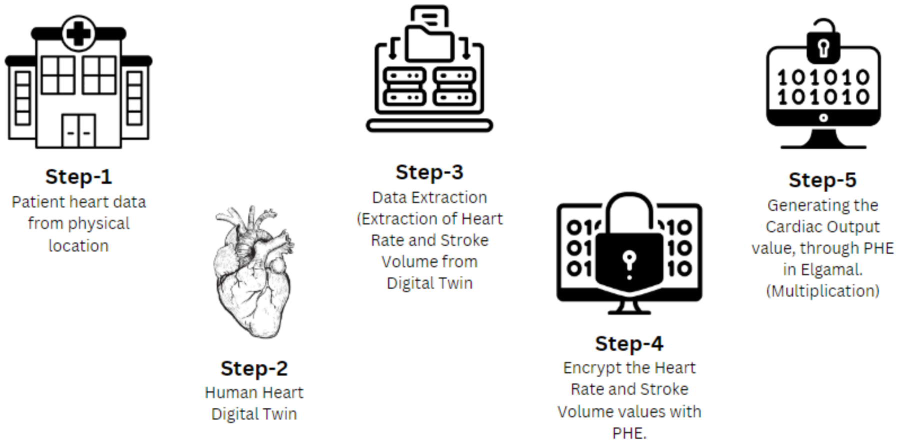
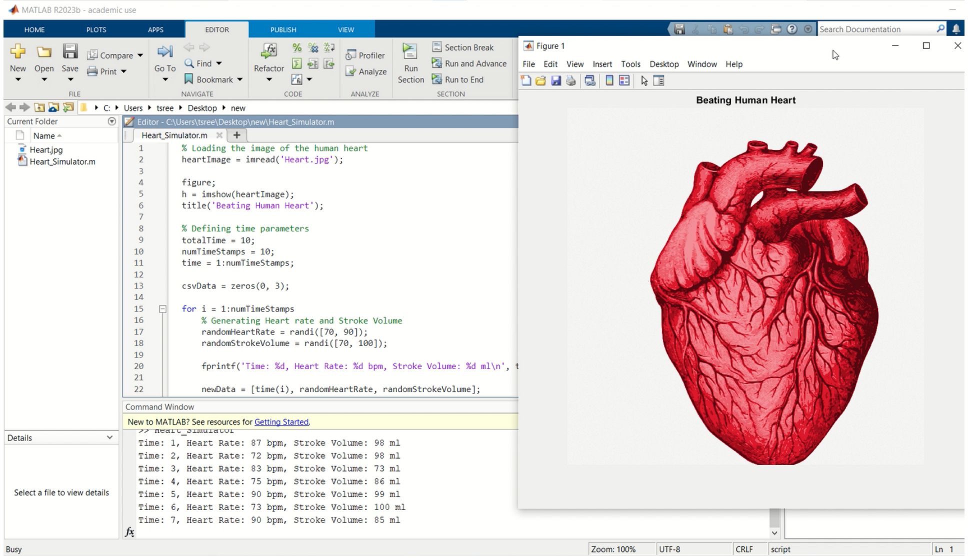

Step-1
Patient heart data from physical location
Step-2
Human Heart Digital Twin
Step-3
Data Extraction (Extraction of Heart Rate and Stroke Volume from Digital Twin
Step-4
Encrypt the Heart Rate and Stroke Volume values with PHE.
Step-5
Generating the Cardiac Output value, through PHE in Elgamal. (Multiplication)

Figure 3: WorkFlow of our Project

The project workflow involves first extracting the patient's cardiac data from the physical location (Hospital). This can be done using various methods, such as electrocardiography (EKG) or echocardiography. We then upload this data to our Digital Twin Simulator in the Matlab software and create a digital replica of the patient's heart in the virtual environment. The Digital Twin then generates the patient's Heart Rate and Stroke Volume. Utilizing Partial Homomorphic Encryption based on the ElGamal algorithm implemented in Python, we then obtain the Cardiac Output value. This represents the output generated from the multiplication of the encrypted Heart Rate and Stroke Volume.



```
% Loading the image of the human heart
heartImage = imread('Heart.jpg');

figure;
h = imshow(heartImage);
title('Beating Human Heart');

% Defining time parameters
totalTime = 10;
numTimeStamps = 10;
time = 1:numTimeStamps;

csvData = zeros(0, 3);

for i = 1:numTimeStamps
    % Generating Heart rate and Stroke Volume
    randomHeartRate = randi([70, 90]);
    randomStrokeVolume = randi([70, 100]);

    fprintf('Time: %d, Heart Rate: %d bpm, Stroke Volume: %d ml\n', t

    newData = [time(i), randomHeartRate, randomStrokeVolume];
```

```
>> Heart_Simulator
Time: 1, Heart Rate: 87 bpm, Stroke Volume: 98 ml
Time: 2, Heart Rate: 72 bpm, Stroke Volume: 98 ml
Time: 3, Heart Rate: 83 bpm, Stroke Volume: 73 ml
Time: 4, Heart Rate: 75 bpm, Stroke Volume: 86 ml
Time: 5, Heart Rate: 90 bpm, Stroke Volume: 99 ml
Time: 6, Heart Rate: 73 bpm, Stroke Volume: 100 ml
Time: 7, Heart Rate: 90 bpm, Stroke Volume: 85 ml
```

Figure 4: Screenshot of the Human Heart Simulator

5.2 Project Challenges and how we overcome them

The major hurdle we faced during the implementation of our project involved finding a suitable digital Twin for the human heart. The ones we discovered were either too complex to comprehend or lacked the data we required. Therefore, we decided to develop our own heart simulation using Matlab. While it presented some challenges, developing our own simulation proved to be the most effective solution, as it allowed us to obtain the desired output from the Digital Twin. This experience provided a valuable learning opportunity.

6 Testing and Experiments

We successfully demonstrated a proof of concept for our proposed solution. We implemented and tested the solution using varying data sizes. Two data sizes were used for testing, with ”Encryption Time” and ”Decryption Time” serving as evaluation metrics. When we used 10 sets of Heart Rate and Stroke Volume values the data was sized to 121 bytes. Encryption took approximately 1.14 milliseconds and decryption took 0.38 milliseconds. When the data was bumped to include 100 sets of Heart Rate and Stroke Volume values, the encryption time went up to 2.67 milliseconds, and decryption took 0.89 milliseconds with a data size of 1157 bytes.

The proposed solution successfully addressed and prevented the potential for any reconnaissance attacks by ensuring that the data cannot be manipulated, tampered with, or accessed. To overcome these challenges, we introduced homomorphic encryption with

Input Data	10 instances (10 values of Heart Rate and Stroke Volume)	100 instances (100 values of Heart Rate and Stroke Volume)
Encryption Time (in milliseconds)	1.14	2.67
Decryption Time (in milliseconds)	0.38	0.89
Test Data (in bytes)	121	1157

Table 2: Testing results

the ElGamal algorithm, which proved to be highly effective. This powerful encryption technique not only kept our data safe from tampering but also ensured that only authorized individuals could access and control it.

7 Future Scope

Looking ahead, there are some exciting possibilities for this project. While our current focus is on multiplying data, we can also perform addition operations going forward by modifying the ElGamal algorithm, as it also possesses additive homomorphic properties. Now that we obtain encrypted, computed data in Layer 2, we can explore ways to protect the other layers of the Digital Twin by leveraging this encrypted data for data processing and manipulation tasks. By displaying this encrypted computed data on Layer 4, we can also address the restricted user data access issue, as we don't need to share the raw data with users who require limited visibility. Real-time data processing will present challenges in terms of speed and efficiency, so we aim to improve the speed and scalability of our system to handle a larger volume of data in real-time.

8 Conclusion

Aiming to address concerns including data tampering, privacy leaks, privilege escalation, and visualization modification, in Digital Twins we introduced a solution utilizing Partial Homomorphic Encryption. This enabled us to perform complex computations on Digital Twins without exposing sensitive data. Our focus was on enhancing the security of Layers 2 and 3 within the Digital Twin environment, and we successfully demonstrated a solution for these security concerns. While further exploration remains, we have effectively overcome initial challenges. Our project is not just a solution; it represents a significant step towards a more secure and reliable Digital Twin ecosystem.

9 Acknowledgment

We thank our professor for the guidance and support provided to us in successfully completing our project.

10 Appendix

The source code for our implementation can be found in this Git Repository linked below. https://github.com/st6626/Enhancing-Digital-Twin-Security-Through-Partial-Homomorphic-Encryption.git

References

[1] C. Gehrmann and M. Gunnarsson, "A digital twin based industrial automation and control system security architecture," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 669–680, 2020.

[2] C. Alcaraz and J. Lopez, "Digital twin: A comprehensive survey of security threats," *IEEE Communications Surveys Tutorials*, vol. 24, no. 3, pp. 1475–1503, 2022.

[3] E. Karaarslan and M. Babiker, "Digital twin security threats and countermeasures: An introduction," in *2021 International Conference on Information Security and Cryptology (ISCTURKEY)*, 2021, pp. 7–11.

[4] Y. Wang, Z. Su, S. Guo, M. Dai, T. H. Luan, and Y. Liu, "A survey on digital twins: Architecture, enabling technologies, security and privacy, and future prospects," *IEEE Internet of Things Journal*, vol. 10, no. 17, pp. 14 965–14 987, 2023.

[5] Y. Tsiounis and M. Yung, "On the security of elgamal based encryption," in *Public Key Cryptography*, H. Imai and Y. Zheng, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 117–134.

[6] S. J. Mohammed and D. B. Taha, "Performance evaluation of rsa, elgamal, and paillier partial homomorphic encryption algorithms," in *2022 International Conference on Computer Science and Software Engineering (CSASE)*, 2022, pp. 89–94.