future internet

MDPI

Article

Applying Detection Leakage on Hybrid Cryptography to Secure Transaction Information in E-Commerce Apps

Mishall Al-Zubaidie * **and Ghanima Sabr Shyaa**

Department of Computer Sciences, Education College for Pure Sciences, University of Thi-Qar, Nasiriyah 64001, Iraq; msc21co3@utq.edu.iq
* Correspondence: mishall_zubaidie@utq.edu.iq; Tel.: +964-61469869029

Abstract: Technology advancements have driven a boost in electronic commerce use in the present day due to an increase in demand processes, regardless of whether goods, products, services, or payments are being bought or sold. Various goods are purchased and sold online by merchants ($\mathcal{M}$)s for large amounts of money. Nonetheless, during the transmission of information via electronic commerce, $\mathcal{M}$s' information may be compromised or attacked. In order to enhance the security of e-commerce transaction data, particularly sensitive $\mathcal{M}$ information, we have devised a protocol that combines the Fernet (FER) algorithm with the ElGamal (ELG) algorithm. Additionally, we have integrated data leakage detection (DLD) technology to verify the integrity of keys, encryptions, and decryptions. The integration of these algorithms ensures that electronic-commerce transactions are both highly secure and efficiently processed. Our analysis of the protocol's security and performance indicates that it outperforms the algorithms used in previous studies, providing superior levels of security and performance.

Keywords: asymmetric cryptography; DLD; e-commerce transactions; Fernet; key identifier; merchant request; probability of data leakage; robust randomization

check for updates

Citation: Al-Zubaidie, M.; Shyaa, G.S. Applying Detection Leakage on Hybrid Cryptography to Secure Transaction Information in E-Commerce Apps. *Future Internet* **2023**, *15*, 262. https://doi.org/10.3390/fi15080262

Academic Editors: Weizhi Meng and Christian D. Jensen

Received: 4 July 2023
Revised: 26 July 2023
Accepted: 26 July 2023
Published: 1 August 2023

1. Introduction

The act of purchasing, offering for sale, or exchanging products, services, and information across different networks of computers/Internet is referred to as electronic commerce (commonly abbreviated to "e-commerce"). E-business encompasses a wider scope than e-commerce and includes various aspects such as customer service, business partnerships, and job openings, among others. E-commerce, a subset of e-business, focuses on commercial activities, but extends beyond simple buying and selling. It involves the use of database or dataset technology, web network technology, email, and other non-computer technologies like systems for goods transportation and online payment options. The term "e-commerce" is often used to collectively describe the electronic marketing of services, goods, and information [1]. Merchants can utilize an electronic system, such as a computer network, television, radio, or the Internet, in order to purchase, sell, and market goods and services. Electronic commerce is becoming increasingly popular around the world as an essential and inevitable practice result of the most recent advances in the services and communication techniques areas. E-commerce is considered one of the most significant applications of Information and Communication Technology (ICT) and is an essential component of the knowledge economy. Nations need to be aware of the importance of e-commerce to keep up with the current environment and advancements driven by its representatives. As e-commerce gains wider acceptance, the concept of the "e-merchant" has emerged, referring to individuals or businesses that primarily use e-commerce applications to fulfill their needs and desires [1,2]. Retaining merchant information and protecting it is very important in e-commerce and online transactions. Access to merchants' information can lead to huge losses that can cause merchants to go bankrupt and even affect the country's economy. The report in [2] indicated that a pharming attack constitutes 43% and an identity theft attack accounts for 33% of types of fraud experienced by merchants in 2023. Also, according to the report in [3], Recorded Future tracked 21 card checker services that utilized 2953 different merchants and 660 different merchant identification numbers (MIDs) for unauthorized card checks.

The security of e-commerce plays a crucial role in preventing undesirable incidents like data leakage and financial losses. Safeguarding merchant information and securing transactions in e-commerce are of utmost importance [4]. The growing adoption of e-commerce has led to an increase in the number of people providing their personal information in various applications. To ensure the security of private data during e-commerce transactions, merchants must prioritize data security to safeguard personal transaction information from hackers. Standard security measures include authenticating merchant information, maintaining transaction data confidentiality, and ensuring data integrity [5]. The protection of information is the main objective of these security measures [6]. Since security attacks are linked to online shopping, encryption is used for e-commerce transactions, and using encryption from public and private encryption technologies, such as Rivest–Shamir–Adleman (RSA), Data Encryption Standard (DES), and TWOFISH, offers a framework that is both extremely safe and productive. However, some weaknesses in these algorithms that affect security transactions: the RSA algorithm is slow, the mathematical operations are large, and the TWOFISH algorithm is more complex in comparison to other old standards such as DES [7]. Personal information must be kept secure to minimize risks [8]. Some of the critical risks and attacks that information faces include camera and double swipe attacks, collusive attacks, dictionary attacks, impersonalization, pharming, smishing, snooping, unfair evaluation, and vishing. These attacks attempt to penetrate transaction information or penetrate personal information by vishing and stealing customers' personal information in e-commerce transaction applications [1]. In the realm of electronic commerce applications, transactional information is susceptible to various types of penetrations or attacks, whether through mobile phones or websites on the Internet. As a result, ensuring security becomes a fundamental and essential task to tackle. In this study, we propose employing encryption algorithms, namely the ElGamal algorithm and the Fernet algorithm, to address this security concern. Our main contribution to this research is as follows:

- We design a robust protocol that achieves lightweight, high-performance encryption operations through the ElGamal algorithm for key generation, and the Fernet algorithm for information encryption and decryption operations.
- We propose the utilization of an information leak detection mechanism in key generation, encryption, and decryption processes to ensure that merchants' information is not exposed.
- We test the performance of our e-commerce application transaction protocol using the Scyther high-security proof tool.

Here are the main points, followed by a succinct elucidation of how the paper is structured: E-commerce and security threats were introduced in Section 1. We outline relevant e-commerce security research in Section 2. The importance of the merchant information in e-commerce transactions is presented in Section 3. Background information about the approaches employed is outlined in Section 4. Methods for the suggested encryption scheme are presented in Section 5. Section 6 explains the proposal's findings and discussion. We quickly summarize our study findings and future directions in Section 7.

2. Literature Review on Commerce Transaction Requests Encryption

Despite the utilization of both symmetric and asymmetric encryption methods in e-commerce applications, significant challenges in terms of security and performance persist with the existing solutions. This section will offer comprehensive evaluations of current research studies related to the subject matter of our study.

Sidik et al. [9] suggested a technique in the one-time pad (OTP) manner's flaw that can be concealed by altering each cipher text in the three pathways used in the three-passes protocol method. To modify the cipher text, a combination of the ElGamal and

RSA algorithms is employed to generate super cipher text. The first and third lines are encrypted using RSA, while the second line undergoes encryption with the ElGamal algorithm. However, this approach presents several issues, including the usage of large initial numbers and complex operations, reliance on multiple keys with different lengths, vulnerability of the one-time pad due to the use of a single key for a single operation, and the susceptibility of the key to being easily cracked. Ali et al. [10] submitted a proposal to develop a trustworthy algorithm for multi-factor authentication for mobile payment systems. In order to increase security when authenticating mobile money, they used a cutting-edge strategy that combined a personal identification number (PIN), an OTP, and a biometric fingerprint. Additionally, they used a quick response (QR) code and biometric fingerprint to validate a mobile money withdrawal. The privacy of the OTP and PIN is enforced by fast identity online (FIDO), which employs a biometric fingerprint and RSA standard public key cryptography in addition to Fernet encryption, to protect a QR code and the data in the datasets. The weaknesses in their proposal include the complexity and large mathematical operations of the RSA algorithm, leading to system slowdowns. Additionally, external conditions like exposure to burns and diseases can alter the fingerprint, impacting both the performance and security of their proposed system.

Tyagi [11] proposed a method to protect data in cloud computing, specifically using image double-level encryption through convolutional neural network (CNN) auto-encoders combined with advanced encryption standard (AES) and Fernet. The process involves processing, encrypting, and decrypting the source images to produce bitmap images as outputs, which users can then decrypt using a key. However, their proposal faces some challenges, such as the double encryption level affecting performance and the potential exposure of data and information to theft and damage when stored in cloud computing. Dong [12] proposed a method that utilizes sensor technology and a smart platform for mining and analyzing e-commerce data. Based on the analysis, a new mobile e-commerce platform was designed, using Jingdong and Taobao as examples. Online evaluation surveys and research were conducted to determine the factors influencing logistics services and customer satisfaction under various logistics distribution models. However, there are some weaknesses in the proposal. Customer satisfaction is influenced by various factors, such as the quality of goods and services, delivery time, speed of delivery, and the attitude of delivery staff. The difference in delivery and delivery time can impact e-commerce platforms. The three-level system construction model increases user–server interaction, but also leads to a huge dynamic page containing both performance and generated data. This complexity poses security risks to the system and makes system development and maintenance challenging.

Abdul Hussien et al. [13] proposed an agent program installed on each customer device to handle security and purchases automatically. The encryption algorithm used strikes a balance between time and complexity, with improvements made to the AES encryption. Preprocessing steps such as zigzag and padding were added, the sub-byte step was removed, and the number of rounds decreased. However, their proposed system has the drawback of significant arithmetic operations, resulting in reduced algorithm speed, increased file memory size, and higher cost. On the other hand, Kota [14] proposed hybrid encryption for data storage in cloud computing. They use AES-GCM, Fernet, AES-CCM, and CHACHA20 POLY1305 algorithms for data security by block. The technique is commonly used for securing key information, with a key size of 128 bits. There are N parts to one file. Each part of the file is encrypted with a special algorithm. All files are encrypted concurrently using two distinct techniques. For the purpose of file decryption, the encryption process is reversed. Their proposal faces certain challenges, such as the use of GCM-AES to encrypt file segments, which requires minimal time and offers the highest throughput for encryption and decryption compared to similar algorithms. However, the process of dividing files into parts and having each part perform a different algorithm leads to extensive and intricate calculations.

Koppaka and Lakshmi [15] proposed a method that utilizes encryption algorithms in hyperchaotic sequences, incorporating the ElGamal algorithm to effectively encrypt outsourced data and reduce computing complexity. They introduced an improved ElGamal cryptosystem (IEC) algorithm, which significantly enhances data security in cloud scenarios by strengthening key pairs through a combination of the classic ElGamal algorithm and pseudorandom sequences for pseudorandom key generation. However, the IEC algorithm has different key lengths, leading to complex operations and negatively impacting system performance and computational complexity. On the other hand, Charles et al. [16] improved the ElGamal encryption–decryption technique to enhance data protection. Their approach involves the use of a newly created private key and a public key for decryption. Encrypted data are decrypted based on a user's request using ResNet-50's nearly 50-layer CNN classifier. Nevertheless, there are concerns about potential attacks on user data containing sensitive information, and the usage of ResNet incurs high costs when dealing with multiple parameters. Ahmed and Ahmed [17] introduced a proposal to employ encryption methods to protect networks and devices connected to each other. The challenge lies in achieving quick and reliable communication among multiple devices without interruptions. Comparing algorithms based on key size, message size, and execution time is crucial [18,19]. Vulnerabilities in the long key RSA algorithm lead to encryption delays and complex operations. Similarly, ECC experiences sluggishness in public key operations and is susceptible to performance-affecting attacks. Parvathi et al. [20] proposed using Fernet/AES with blockchain technology in the food supply chain to ensure secure transactions between farmers and consumers/buyers. However, this approach faces issues as data processing for each purchase and sale order takes time, affecting system performance and causing delays in orders and potential damage to goods.

3. Importance of the E-Merchant in E-Commerce Transactions

E-merchant is known as a commercial transaction, conducted electronically in facilitating both marketing and stalking operations anywhere, anytime and with whoever participates in the transaction over the Internet. This adaptability is what attracts customers, and merchants can increase the sales of their products by partnering with multiple websites. Customers can buy goods and/or services directly from online retailers. Merchants deal on a day-to-day basis on their websites. They sell goods and services daily to customers for a fee, and often have coupons on the website. Online market merchants offer to sell goods or services that will be sold in online stores via online shopping malls by uploading data or information [2]. E-commerce provides a number of merchants and platforms via the Internet, and the merchant is responsible for the quality of the product and its price, and the quality of sales. E-merchant platforms play a crucial role in the financial gains of platform merchants by managing user interactions, curating content, and imposing transparent and flexible management limits. Large online marketplaces like eBay, Amazon, and Alibaba have a significant impact on content selection, categorization, and display. However, the quality of goods sold by merchants on these platforms may be questionable. Some sellers resort to deceptive advertising, leading to the selling of subpar goods in e-commerce transactions [21].

The sales volume is influenced by pricing, which is set by the merchants. Consumers interact with merchants to inquire about product prices. Social media advertising and consumer engagement for products on social platforms can boost merchant sales and generate commissions. Merchants invest in search engine marketing to promote suggested products to consumers [22]. The availability of numerous deals and opportunities to make purchases through Internet sales has increased significantly. (1) Consumers and e-merchants interact online through a server rented by the e-merchant from an Internet service provider (ISP). (2) All online transactions include terms of use and terms of sale, which are typically posted on the e-merchant's website. Interested customers must click the accept button to agree to these terms. (3) By clicking the accept button, customers electronically bind themselves to the contract with the e-merchant. (4) The payment process involves two intermediary banks—the commercial bank of the acquiring party and the bank's issuing bank. The client mechanism authorizes the e-issuing customer's bank to make payments to the purchasing merchant's bank on behalf of the customers for the cost of the items. (5) Once the payment process is complete, the e-merchant fulfills its obligations by delivering the items according to the agreed timing and product specifications [23]. The importance of the e-merchant in e-commerce to protect the security of information, customer data, and product data is crucial in electronic commerce to prevent theft and ensure data protection. To achieve this, built-in encryption techniques are used, providing high data security and promoting products effectively. The process involves implementing encryption and authentication protocols to safeguard sensitive information exchanged between the merchant and goods suppliers, as well as between the merchant and customers. Transactions in e-commerce involve sensitive data such as names, addresses, mobile numbers, and banking information, making it a target for attackers. Protecting this information is essential for the security of electronic commerce, as the merchant acts as a link between suppliers and customers.

4. Basic Concepts about Applied Cryptography Mechanisms

In this section, we will provide the basic details of the algorithms adopted in the current research.

4.1. ElGamal Algorithm

ElGamal cryptography, known for being one of the earliest and well-known public key encryption methods, gained popularity in the 1980s due to its effectiveness and lack of patent restrictions [24]. It relies on the use of random integers during key generation, making it a relatively secure approach. The ElGamal algorithm is recognized for its simplicity and efficiency in various cryptographic operations, providing protection against threats in computer networks, including e-apps, online websites, and cryptanalyst attacks [24,25]. As an asymmetric cryptography method, the ElGamal algorithm involves both public and private keys in the encryption and decryption processes. Its security is based on the complexity of discrete logarithms. The key ElGamal algorithm is created using the following: p = primes and $y = gx\ mod\ p$ stands for random numbers and $g < p$. Conditions: x = random numbers, $x < p$. The following is how the text is encrypted: $a = gk\ mod\ p$, where k denotes a random number decrypting text involves the following: $m = b * a(p - 1 - x)\ mod\ p$ [25]. The ElGamal cryptosystem is a very effective application of the Diffie–Hellman algorithm; the cipher text for a particular message m is not repeated due to randomness in the enciphering process. The distinguishing characteristic of the ElGamal algorithm's encryption and decryption is the use of the residual obtained when a large number is divided by a prime number. Since there are countless ways to divide the number, it becomes exceedingly challenging to identify the original unique combination of factors that produced that specific residual [26]. The key size utilized by ElGamal's approach will eventually be utilized to calculate the positive prime number p and the integer q, which is the primitive root of p. For instance, a low parameter will be used to increase accuracy and make calculations simpler such as a key length of 1024-bit.

4.2. Fernet Algorithm

Fernet is a cryptography technique that offers a straightforward mechanism for authenticating and encrypting data employing symmetric AES-128 in CBC (Cipher Block Chaining) mode with public-key cryptography Standards (PKCS7) padding to allow various lengths of 128 bits (16 bytes) [27]. The Fernet algorithm ensures that the data are encrypted, making it impossible for them to be modified or decoded without the key. Fernet is a recent symmetric encryption and decryption system that ensures message authenticity. This allows recipients to detect any alterations from the original transmission. To avoid common mistakes made by inexperienced developers, Fernet provides a secure key generation technique, utilizing efficient encryption (AES), and enhancing security through random "salt" value generation with CBS mode and PKCS7 padding. Fernet supports both symmetric and private keys, where a single key is used for encryption and decryption. However, it may have limitations with large files, requiring a single memory load of the entire buffer [28].

The Fernet keys make sure that an encrypted template file cannot be revealed or read without the secret key, making it challenging for an attacker ($\mathcal{A}$) to circumvent or access the database server [29]. A symmetric key approach is used to ensure that the encrypted transmission cannot be changed, brute-forced, or decoded without the key. To enhance security, every character in the key undergoes base64 URL-safe encoding, which includes substituting reserved, illegible, or non-ASCII characters. It makes sure that the keys are handled correctly and that no mistakes happen that an $\mathcal{A}$ would try to take advantage of. PKCS7 padding and Fernet both utilize the 128-bit cipher block chaining (CBC) mode of the advanced encryption standard (AES). PKCS7 padding is employed to fill in the vacant bits, ensuring that the cipher remains in multiples of 128 bits. For password key usage, HMAC (hash-based message authentication code) is employed, serving two functions: confirming the authenticity and integrity of a message. To enhance security, HMAC is combined with a straightforward 256-bit hashing algorithm (SHA256) [30]. The encryption procedure performs SubBytes(), RowsShift(), AddRoundKey() and MixColumns() operations, while the decryption procedure performs operations in reverse to obtain the plaintext. Higher levels of trust, security, authenticity, privacy, user authorization, non-repudiation, information integrity, and secrecy are provided by Fernet. This algorithm guards against attacks including impersonation, shoulder surfing, pharming, identity theft, and dictionary threats [10].

4.3. Data Leakage Detection Technique

Data leakage detection (DLD) is a well-known process designed to identify and prevent the unauthorized disclosure of sensitive information. DLD systems are specialized tools that monitor and safeguard personal data, identifying instances of data misuse and identifying the source of the leak. Data leakage can occur both intentionally and accidentally, where private or confidential information is shared with unauthorized parties. Figure 1 illustrates the data leakage process in e-commerce applications. In the context of business operations, an entity's sensitive data may need to be shared with various stakeholders such as clients, employees, and business partners, whether they are inside or outside the organization's premises. However, there is a risk that the recipients may misuse or unintentionally disclose this information to unauthorized third parties [31]. Data leakage presents a significant challenge in the modern business landscape as safeguarding data from unauthorized access is crucial [1]. The uncontrolled leakage of data can expose businesses to various risks, and once sensitive information leaves the organization's domain, it puts the company in serious jeopardy. Even a single attack on a company can impact a large number of customers, particularly merchants, along with their distinct data [32]. Data leakage has become a major problem for organizations, and it often goes unnoticed as it originates from diverse sources that are difficult for most individuals to identify. The expanding global nature of businesses has further complicated the situation, as sensitive information can be electronically transferred using various technological devices like USB keys, spreadsheets, and web pages [1]. This underscores the increasing importance of addressing data security, especially when one client needs to transmit information from one nation to another through intermediaries. Failure to address data security could lead to severe financial losses for the company if the data are released. The process for detecting data leaks is

1. The distributor enters their login information.
2. The distributor enters the data (for instance, text files) into the database.
3. After logging into the system, the agent requests a specific file, or the distributor uploads all files for the agents appropriately, along with the private key.
4. The distributor delivers the desired file to the requested agents, who then add some fictitious objects.

5. According to his/her demands (explicit requests or sample requests), agents will download the files.
6. The distributor will search for the leaked data and locate the file if any agents (fake agents) release the information to a third party [32].

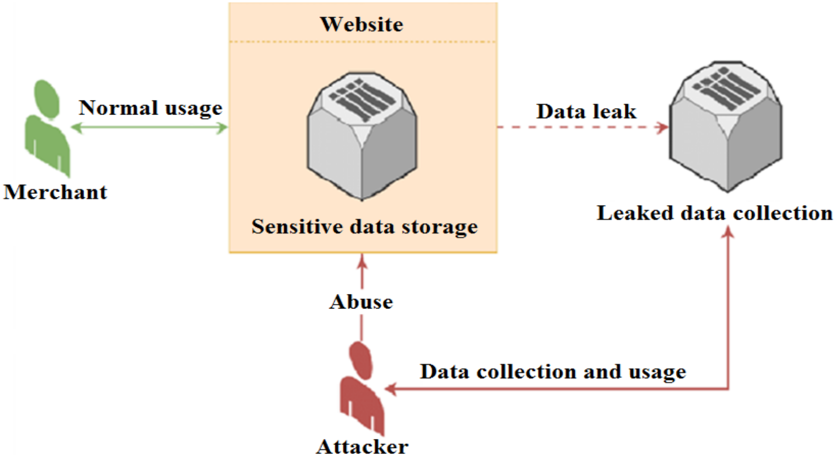


Figure 1. Data leakage detection technique.

5. Proposed Protocol to Secure E-Commerce Transactions

The proliferation of wireless communication networks, credit cards, smartphones, and the continuous growth of e-commerce have led to increased product sales and delivery. E-commerce involves a complex system comprising various elements, including the Internet, online shopping websites, servers, payment methods, product delivery, and customers. However, the transactional information in e-commerce faces potential breaches and threats, emphasizing the need to protect data privacy. To ensure information security, we propose a high-performance and secure protocol. The proposed system's general model includes customers, merchants, payment gateway operations, payment methods, banking services, and online sales operations. Our focus lies in securing the order processes between the merchant and their organization. To achieve this, we employ both symmetric and asymmetric encryption algorithms and leverage DLD technology to identify potential data leaks in the e-commerce order flow during key generation, encryption, and decryption. Figure 2 illustrates the general structure of our proposed system, while Figure 3 depicts the methodology of our protocol.

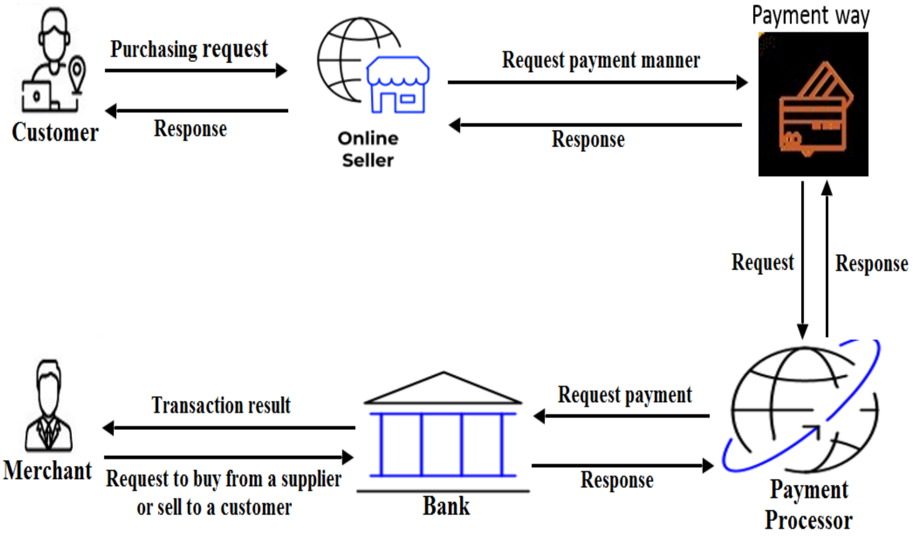


Figure 2. Our proposed system.

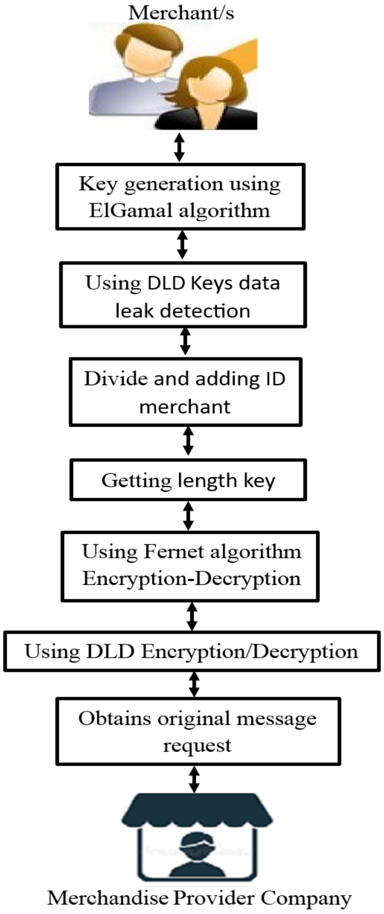

Merchant/s
Key generation using ElGamal algorithm
Using DLD Keys data leak detection
Divide and adding ID merchant
Getting length key
Using Fernet algorithm Encryption-Decryption
Using DLD Encryption/Decryption
Obtains original message request
Merchandise Provider Company

Figure 3. Diagram of protocol methodology.

5.1. Employing Secret-Key and Public-Key Encryptions

The methods employed in our proposed protocol, ElGamal (ELG), Fernet (FER), and DLD, are described in this section.

5.1.1. ElGamal

In our protocol, we implement an asymmetric algorithm that involves two keys: a public key and a private key. These keys are utilized to generate large and random keys, enhancing the security of e-commerce transactions. We choose the ELG algorithm due to its high efficiency in generating random keys of various lengths and sizes, such as 768, 1024, 2048, 3072, 4096, 7680, and 15,360 bits. In our proposed protocol, we use a key size of 1024 bits. To generate the secret key, we divide the key obtained from the ElG algorithm using the XOR process, resulting in a secret key size of 256 bits. This secret key is employed for encryption and decryption operations using the FER algorithm. The ElG algorithm is also used to generate the public and private keys. The key generation process is as follows:

- Select a large prime number at random q;
- Select a random number g, which referred a random multiplicative to as a generator component;
- Select a third number at random K_r from $1 \ldots q-1$ as the private key;
- Calculate y by using the formula: $y = gK_r \; mod \; q$ as the public key;
- K_r should be kept secret as a private key, q, g and y are published as public key (K_u).

We use data leakage detection technology during the process of creating the keys. Next, we divide the K_u into four sub-keys, and with the addition of the merchant ID number for each sub-key and the XOR operation for each sub-key, we derive the public key encryption and decryption using FER. Figure 3 illustrates the methodology of the system.

5.1.2. Fernet

The Fernet algorithm encrypts and decrypts product sales, delivery orders, and payment gateway paths between the merchant ($\mathcal{M}$) and trusted server (TS) device with a key length of 256 and salt values to increase randomness and provide more security. In our protocol, we achieve a balance between lightweight implementation and high-security operations by utilizing the FER algorithm. This algorithm ensures robust security for merchant information and effectively safeguards e-commerce transactions from hacking and tampering. Figure 4 illustrates the architecture of the Fernet algorithm in our proposed protocol.

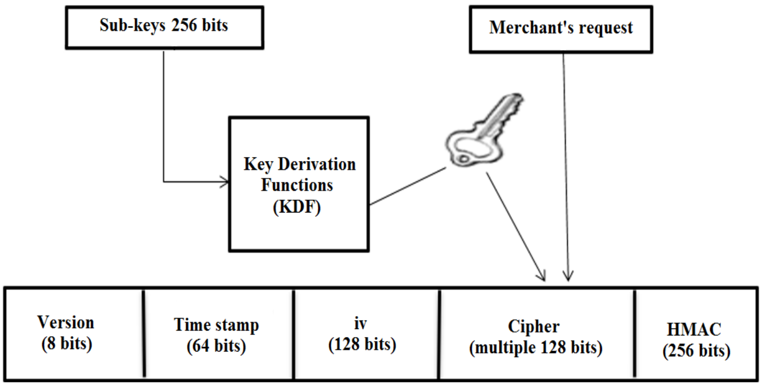


Figure 4. Our FER architecture.

5.1.3. DLD

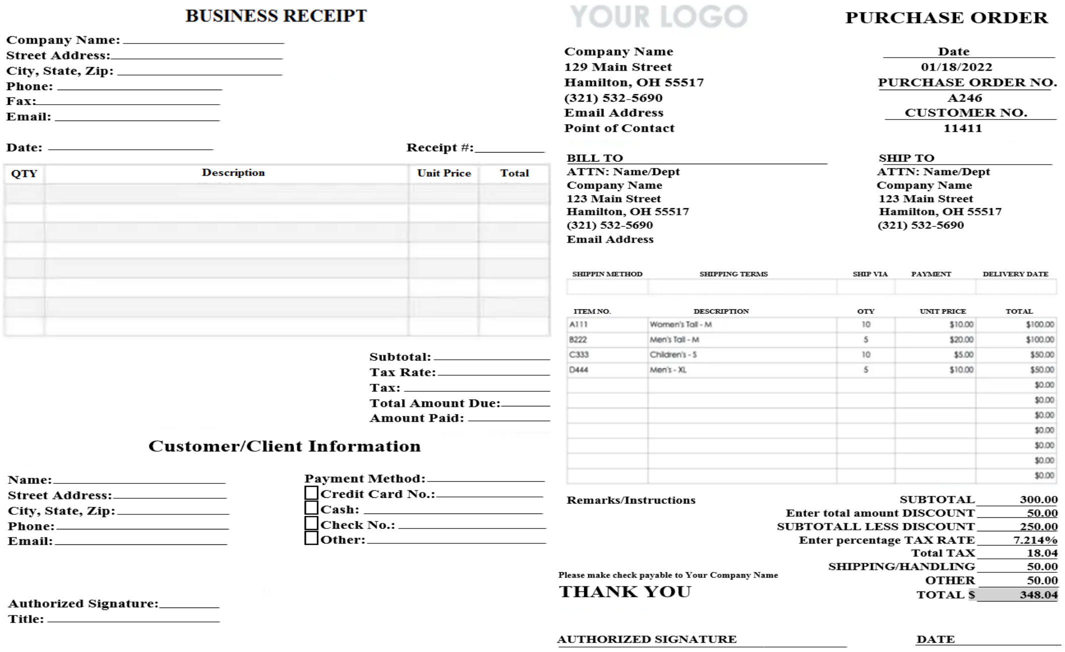
Information or data in e-commerce transactions can be vulnerable to hacking and unauthorized access, leading to potential data leakage. Protecting sensitive information during the transmission is crucial. When a merchant sends a product request to a company's website, it is essential to safeguard the merchant's data. Any mishandling of the order data or involvement of third-party agents in data transmission can result in data leakage. To address this, our protocol incorporates technology to detect data leakage during key generation, encryption, and decryption processes. We utilize dummy objects that mimic real agents to trace the source of leaks and identify responsible parties.

Give GM represents the group of $\mathcal{M}$s and GD denotes the company's group dataset. A is the agent, and GE_A refers to the agent's guilty event. It should calculate an agent's P_i probability of being a guilty agent. When given the leaked data LD, the probability is indicated by $P_i = GE_A|LD$. We suppose that $\forall\ D_i \in LD$, D represents sensitive data from the database, and $i = \{v_1, v_2 \ldots, v_n\}$. There can only be two conceivable outcomes. The first outcome is that any single agent from the set of $S(D_i) = A|D_i \in X_j$ has leaked D_i to target t, where $S(D_i)$ signifies the set of agents who have D_i in their allocated database $X_j\ \forall\ j = \{1, 2, \ldots, n\}$. Alternatively, the target t obtained the data D_i by chance or some other means without the assistance of any agent A. The probability of leaking any data item D_i from the leaked database LD, i.e., P_i (leak D_i to LD) equals $\forall\ A \in S(D_i)$ if leaked by any agent $A \in S(D_i)$, otherwise P_i (leak D_i to LD) equals $alpha$ if acquired by the target t. We contend that A's resolution to leak any data D_i is independent of the leaking of other data D_i, $\forall\ D_i$, $D_i \neq D_{i_{new}}$. $P_i(GE_A|LD)$ of the agent A to be a guilty agent GE_A is calculated, as shown in the following equation.

$$P_i(GE_A|LD) = 1 - \prod_{D_i \in LD \cap X_j} (1 - (1 - \alpha)/C(D_i)) \tag{1}$$

5.2. Purchase and Payment Requests

The FER algorithm is employed to carry out cryptographic tasks on a collection of cryptographic orders, including buy and payment orders. Figure 5 illustrates the different types of requests. Information from order requests, such as purchase or payment orders, is encrypted based on the order type. The order formats used in this study are real e-commerce orders, such as purchase and payment requests, obtained from publicly available databases [33] on the Internet, without specific party data like company, institution, or personal information. To facilitate the encryption and decryption process of purchase and payment orders, we introduced some non-real user data and personal information. Before sending an order from a merchant to a server, it is encrypted using the FER technique, ensuring the safeguarding of each merchant's personal information.



BUSINESS RECEIPT

Company Name:
Street Address:
City, State, Zip:
Phone:
Fax:
Email:

Date: Receipt #:

QTY	Description	Unit Price	Total

Subtotal:
Tax Rate:
Tax:
Total Amount Due:
Amount Paid:

Customer/Client Information

Name:
Street Address:
City, State, Zip:
Phone:
Email:

Payment Method:
□ Credit Card No.:
□ Cash:
□ Check No.:
□ Other:

Authorized Signature:
Title:

YOUR LOGO

PURCHASE ORDER

Company Name
129 Main Street
Hamilton, OH 55517
(321) 532-5690
Email Address
Point of Contact

Date
01/18/2022
PURCHASE ORDER NO.
A246
CUSTOMER NO.
11411

BILL TO
ATTN: Name/Dept
Company Name
123 Main Street
Hamilton, OH 55517
(321) 532-5690
Email Address

SHIP TO
ATTN: Name/Dept
Company Name
123 Main Street
Hamilton, OH 55517
(321) 532-5690

SHIPPIN METHOD	SHIPPING TERMS	SHIP VIA	PAYMENT	DELIVERY DATE

ITEM NO.	DESCRIPTION	QTY	UNIT PRICE	TOTAL
A111	Women's Tall - M	10	$10.00	$100.00
B222	Men's Tall - M	5	$20.00	$100.00
C333	Children's - S	10	$5.00	$50.00
D444	Men's - XL	5	$10.00	$50.00
				$0.00
				$0.00
				$0.00
				$0.00
				$0.00
				$0.00
				$0.00

Remarks/Instructions

SUBTOTAL	300.00
Enter total amount DISCOUNT	50.00
SUBTOTALL LESS DISCOUNT	250.00
Enter percentage TAX RATE	7.214%
Total TAX	18.04
SHIPPING/HANDLING	50.00
OTHER	50.00
TOTAL $	348.04

Please make check payable to Your Company Name
THANK YOU

AUTHORIZED SIGNATURE DATE

(a) Pament request **(b) Purchase order**

Figure 5. Purchase and payment requests.

5.3. Procedures of Proposed Protocol

In building our protocol, we depend on the mechanism of encrypting and decrypting transaction requests. This is accomplished by employing symmetric and asymmetric encryption techniques. The FER is used to encrypt security parameters and request information. We generally enumerate the steps for the proposed protocol.

1. Using an ELG asymmetric encryption technique, our protocol generates large random keys that are both private and public. The random keys 1024-bit are then divided into 256-bit chunks to accommodate the keys of FER algorithm.
2. This public key is used to encrypt e-commerce requests between $\mathcal{M}$ and TS. When e-commerce requests are transferred from TS to $\mathcal{M}$ or vice versa, our protocol employs strong encryption with high encryption randomness, thus making it difficult to hack.
3. The FER and ELG keys will be used to decrypt the required information. $\mathcal{M}/TS$ device receives a decrypted message that is intractable to perforate by a hacker.
4. We employ DLD technology to safeguard information, particularly $\mathcal{M}$ information, against leakage throughout the key generation, encryption, as well as decryption procedures.
5. Concealing security parameters and request information on devices of networks, particularly keys, is critical in situations of device hacking.

5.3.1. Key Generation Procedure

1. We generate 1024-bit K_u public and K_r private keys using parameters q and g were mentioned by the ELG algorithm in Section 5.1.1;
2. We divide the Ku public key into parts k_1, k_2, k_3 and k_4 with a key size of 256 bits that fits the key size of the FER algorithm, which is 256 bits;

3. We perform XOR operations on keys such as $k_5 = k_1 \oplus k_2$ and $k_6 = k_3 \oplus k_4$, following which we obtain the final key, $FK = k_5 \oplus k_6$;
4. We add an ID to each final key and perform the $FK_i = FK \oplus K_{ID}$ operation;
5. We use DLD to process the leakage probability (P_i) of a (D_i) data group from the guilty party to a group of $\mathcal{M}$s' agents (GM_i);
6. We hide the keys of score (sco)=$PW \oplus K_r$ and $Fk_e = Fk_i \oplus sco \oplus PW$. In the case of the following connection, we do not generate keys. However, we change the ID for each key (K_{ID}). This ID is associated with the key, and not the user (see Algorithm 1).

Algorithm 1 Keys generation procedure.

```
Input: $q$, $g$ values and $PW$, $t$-threshold
Output: $K_r$ and $K_u$ keys with a 256-bit length
1: Using ELG to generate $K_u$ and $K_r$ with 1024-bit length
2: Dividing $K_u$ into four parts with a 256-bit length
3: Four sub-keys: $k_1$, $k_2$, $k_3$ and $k_4$
4: Applying $\oplus$ with sub-keys
5: Obtaining $Fk_i$ with 256-bit length
6: Adding $ID$ for each key, Computing $FK_i \longleftarrow FK \oplus ID$
7: Computing $P_i \longleftarrow GM_i/D_i$
8: If $P_i(D_i) > t$ Declare as Info leakage
9: Else repeat step 7
10: Protecting $K_r$ and $FK_i$ on the device
11: Computing $sco \longleftarrow PW \oplus K_r$
12: Storing $FK_e \longleftarrow FK_i \oplus sco \oplus PW$
13: Next connection go to step 4 with changing $Key - ID$
```

5.3.2. Encryption Procedure

In our protocol, we use PW, K_r, K_u, R_i (sell or buy orders) and t-threshold in encrypting $\mathcal{M}$ transactions, but before performing the encryption process, the stored K_r and FK_i keys should be extracted as in Algorithm 2 from steps 1 and 2. We extract the keys and then we perform the encryption operation using the FK_i key, which is a FER 256 key and a random increase salt ($salt_i$) in order to provide more information security. Our protocol encrypts through $ER_i = En(FK_i||R_i||salt_i||N_{ID} \oplus U_{ID})$, where En represents the encryption process, N_{ID} denotes the network's ID and U_{ID} signifies the user's ID; then we use DLD for leak detection. This happens by checking the leakage of the ciphertext $P_i = ER_i/D_i$. Then, we compare the threshold (t) with the probability of leaking the ciphertext after it is sent over the network connection to the receiving end. We send an encrypted text to the recipient and hide FK_i and K_r with FK_e so that it is protected and hard-to-hack information.

Algorithm 2 Encryption procedure.

```
Input: $K_r$, $K_u$ keys with 256-bit, $PW$ and $R_i$, $t$-threshold
Output: $ER_i$ and $P_i$
1: Extracting $K_r \longleftarrow PW \oplus sco$
2: Extracting $FK_i \longleftarrow FK_e \oplus sco \oplus PW$
3: Using $FK_i$ 256-bit with Ferent encryption
4: Encrypting $ER_i \longleftarrow En(R_i||FK_i||salt_i||N_{ID} \oplus U_{ID})$
5: $P_i \longleftarrow ER_i/D_i$
6: If $P_i > t$ Declare as Info leakage
7: Else repeat step 5
8: Storing $FK_e \longleftarrow FK_i \oplus sco \oplus PW$
9: Storing connection order on the sender side
```

5.3.3. Decryption Procedure

It is a reverse process for an encryption process; public and private keys are used for decryption, and received requests are decrypted upon reaching the recipient. Our protocol decrypts the requests as in Algorithm 3, and we extract the keys from steps 1 and 2 up to $DR_i = De(ER_i)$, where De represents the decryption process, and the decryption process is completed. In our protocol, we use DLD technology to determine the data leakage in the text after decoding and calculating the probability (P_i) and then compare it with the threshold (t) to validate/detect the plaintext of the requests on the receiver device. Then, plaintext should conceal (C_i) using PW and Fk_i through performing an XOR operation with the text $C_i = R_i \oplus PW \oplus Fk_i$ saved to be archived in datasets. Similar to Algorithm 2, Algorithm 3 stores keys anonymously.

Algorithm 3 Decryption procedure.

```
Input: $K_u$ keys with 256-bit, $PW$ and $ER_i$, $t$-threshold
Output: $DR_i$
1: Extracting $K_r \longleftarrow PW \oplus sco$
2: Extracting $FK_i \longleftarrow FK_e \oplus sco \oplus PW$
3: Using $FK_i$ 256-bit with FER decryption
4: Using $ER_i \longleftarrow En(R_i||FK_i||salt_i||N_{ID} \oplus U_{ID})$
5: Decrypting $DR_i \longleftarrow De(ER_i)$
6: $P_i \longleftarrow DR_i/D_i$
7: If $P_i > t$ Declare as Info leakage
8: Else repeat step 6
9: Saving $R_i$ in the dataset
10: Storing $C_i \longleftarrow R_i \oplus PW \oplus Fk_i$
11: Storing $Fk_e \longleftarrow Fk_i \oplus sco \oplus PW$
12: Storing connection order on the receiver side
```

6. Analysis of Proposed E-Commerce Apps' Reliability and Effectiveness

This section will address protection analysis by verifying our protocol's capacity to prevent e-commerce threats. The Scyther tool is then utilized to validate the security of our protocol in practice.

6.1. Security Examination of a Variety of E-Commerce Threats

The following is a summary of the threats analysis:

6.1.1. Camera and Double Swipe

The attacker ($\mathcal{A}$) tries to use the camera to capture the PW/PIN or payment card information, which is recorded when the customer enters private information. Sometimes, the simplest strategy needs a conspirator $\mathcal{M}$. Prior to entering the card into the legitimate device, the merchant strategically positions a camera to capture the PW/PIN pad, and then discreetly swipes the card through their own equipment. Fortunately, even if the $\mathcal{A}$ obtains PW/PIN, they will not be able to complete the purchase or sale order in e-commerce applications as our protocol does not solely depend upon PW because there are a set of security parameters such as FK_i, K_u, K_r and K_{ID}, which the $\mathcal{A}$ is unaware of, which prevent the completion of the authentication process in TS. Furthermore, our protocol takes advantage of DLD technology to detect information leaks such as PW/PIN at TS. If $P_i(DR_i) > t$, then TS reports an information leakage. Thus, the security parameters in our protocol easily prevent this attack.

6.1.2. Collusive Attack

The $\mathcal{A}$ tries to agree with $\mathcal{M}$ or companies in order to raise prices, reduce production, or seize payment or buying and selling operations in a business. It accomplishes this through hack keys (K_u and K_r) and a PW or obtains the secret keys from the collusive legitimate $\mathcal{M}$. In our protocol, we generate asymmetric K_u and K_r keys, which means that if an $\mathcal{A}$ obtains the K_u and K_r keys for a specific $\mathcal{M}$, he/she cannot use those keys to decrypt other $\mathcal{M}$ requests. As a result, our protocol is resistant to this attack.

6.1.3. Dictionary Attack

This threat is a methodical approach to password guessing that uses a large number of common words and their straightforward variants. $\mathcal{A}$s employ huge lists of the most popular names of networks, e-commerce goods, brands, $\mathcal{M}$s, databases fictitious characters, or even simple phrases straight out of a dictionary. The $\mathcal{A}$ accesses the passwords in the dictionary to penetrate transaction information regardless of whether it is an online purchase, sale or payment information. Frequent use of these words can guess PWs by the $\mathcal{A}$. Firstly, in our protocol, PW is not explicitly sent by e-commerce applications. Secondly, it is very difficult to guess PW in our protocol because it is hidden in the process $FK_e = FK_i \oplus sco \oplus PW$ (Algorithm 2). Thirdly, we use the FER algorithm to encrypt all $\mathcal{M}$ requests with highly random keys generated by the ELG algorithm. In this manner, our protocol prevents dictionary attacks.

6.1.4. Impersonalization

The malicious $\mathcal{A}$ pretends to be another $\mathcal{M}$ or company and is socially constituted to obtain, collect information or gain access to an e-commerce app, company, system or organization, or hack sale, purchase and payment information in e-commerce apps. They may use some legitimate information to hack e-commerce applications. In this regard, our protocol uses different parameters like N_{ID}, U_{ID} and K_{ID} to prevent this attack from being executed. Also, the $\mathcal{A}$ does not know that FK_i comprises a combination of keys K_1, K_2, K_3 and K_4 mixed with K_{ID} and is not explicitly sent over the network. These parameters support cryptographic randomness and prevent an $\mathcal{A}$ from performing an impersonalization attack on our protocol.

6.1.5. Pharming

In this threat, the $\mathcal{A}$ employs an online fraud technique through the use of harmful code in order to steer victims to counterfeit apps/websites in an effort to obtain their personal information and login credentials in e-commerce apps. An $\mathcal{A}$ attempts to implant malicious software/requests on a $\mathcal{M}$'s device or TS. In our protocol, TS is reliable and impervious to attacks. Also, all requests are encrypted with the FER algorithm and none of the security parameters are explicitly transmitted. If the $\mathcal{A}$ sends malicious code to TS, it will be outright rejected. Also, if the $\mathcal{A}$ sends malicious code to the $\mathcal{M}$ device and then compromises it, they will not obtain any security parameters that are explicitly stored on the $\mathcal{M}$ device. Therefore, our protocol is capable of blocking this attack.

6.1.6. Smishing

Using a persuasive text request, an $\mathcal{A}$ can persuade their intended victims such as $\mathcal{M}$ to open a link, provide the $\mathcal{A}$ their personal information, or download dangerous software to $\mathcal{M}$s' devices. The $\mathcal{A}$ attempts to gain access to $\mathcal{M}$, TS, organization, or company information in e-commerce applications such as $\mathcal{M}$ identities such as PW and K_{ID}, account information, credit card details, personal information, and payment method in e-commerce transactions. In our protocol, $\mathcal{M}$ identifiers and information (PW and K_{ID}) are not stored on the $\mathcal{M}$ device and are not explicitly transmitted to the receiver. In addition, our applications do not respond to links sent in requests. It is an onerous task to hack information; therefore, our protocol successfully fends off an attack.

6.1.7. Snooping

An unauthorized $\mathcal{M}$ or hacker tries to access the e-commerce company's requests/data, its institution, or a group of legitimate $\mathcal{M}$s. Snooping involves monitoring a message sent through an e-commerce app/program or email to remotely monitor activities on a network or host hacking to capture data like PW, U_{ID}, usernames, addresses, etc. It also includes the interception of data transmission and communication to collect information via network traffic for analysis. Our protocol uses a strong encryption algorithm (FER) with a key length of 256 bits that prevents this attack from analyzing e-commerce requests and accessing security parameters such as K_u and K_r.

6.1.8. Unfair Evaluation Attack

An $\mathcal{A}$ or the rater is intentionally harmful, the reputation worth of transactional partners in e-commerce is unfairly assessed, and it also tries to penetrate the information of orders and information of $\mathcal{M}$s, such as the transaction number of the payment process, the time of delivery of goods, the quality of products and goods, as well as the safeguarding of this information. We have designed in our protocol large random keys with the size of 1024 bits by ELG, mixed up the keys with K_{ID} to support randomness, and used FER 256 bits to encrypt the information. This, in turn, increases randomness and prevents the $\mathcal{A}$ hacking the $\mathcal{M}$'s information by unfair evaluation attack.

6.1.9. Vishing

By obtaining personally identifiable information, the $\mathcal{A}$ intends to provide fraud access requests to a $\mathcal{M}$'s account. The $\mathcal{A}$ attempts to compromise information related to orders, sales, purchases, and payment processes while this information is being transmitted between $\mathcal{M}$, TS, or another party. Our protocol uses a set of security parameters such as FK_i and sco to achieve robust authentication at the receiver. Also, the attack is prevented by encrypting the FK_i and sco upon being sent across the network. Using the FER algorithm and with a key of 256 bits allows our protocol to provide high security for hard-to-hack e-commerce requests by this attack.

Table 1 provides a comparison between our protocol and modern authentication protocols in repelling various attacks within our research field.

Table 1. Comparison of attacks prevention among encryption protocols.

Attack	[34] 2018	[35] 2020	[9] 2021	[36] 2021	[28] 2022	[37] 2022	[38] 2022	[1] 2022	[39] 2022	[40] 2022	[41] 2023	Proposed Protocol
Camera and double swipe						✓				✓	✓	✓
Collusive							✓					✓
Dictionary			✓		✓			✓				✓
Impersonalization	✓	✓		✓						✓	✓	✓
Pharming	✓								✓			✓
Smishing	✓								✓	✓		✓
Snooping										✓		✓
Unfair evaluation		✓		✓			✓					✓
Vishing	✓								✓	✓		✓

6.2. *Security Analysis Using Scyther*

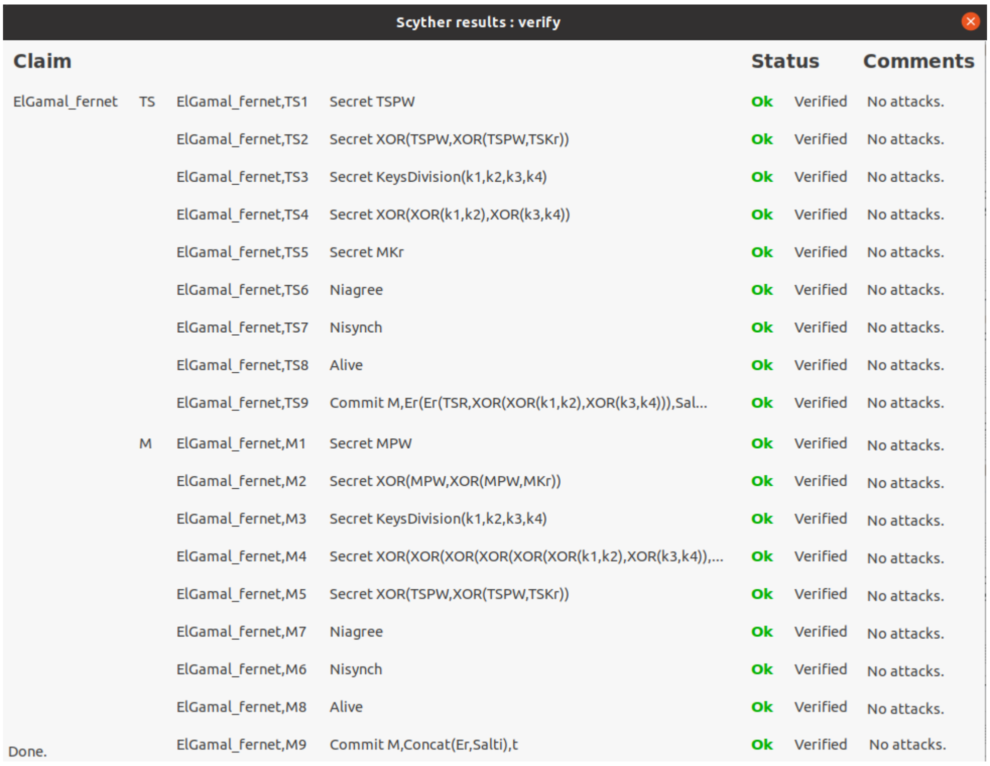
We make use of Scyther, a powerful tool for validating cryptographic protocols. This tool boasts advanced capabilities and is at the forefront of verification speed and attack tracking. It efficiently verifies most protocols for any number of sessions, and all identified attacks are genuine attacks on the model without employing approximation techniques [24]. Users can employ Scyther for attack detection or perform unrestricted verification. Scyther stands out among other protocol analysis tools due to its ability to combine the strengths of theorem proving or abstraction-based approaches (unbounded verification) and model-checking methods (identifying attacks, termination). Moreover, Scyther offers innovative features not found in other tools, such as complete characterization and attack selection. It can be used via the command-line interface, as a backend for analysis programs using Python interface functions, or through the graphical user interface. Scyther uses the analysis of security requirements for a variety of protocols, detecting attacks on information, and verifying the authentication/confidentiality of this information, whether buying and selling operations, payment operations, or sending and receiving operations between a $\mathcal{M}$ and TS, or between companies or institutions. This tool verifies some of the authentication information through security requirement properties, such as Aliveness, Nisynch, Niagree, and Weakagree.

6.2.1. Description of Scyther with Proposed E-Commerce Protocol

We utilize the Scyther tool to assess the effectiveness of our proposed protocol. To prepare our protocol roles for analysis, we employed the Security Protocol Description Language (SPDL) within the Scyther tool. Here, we use a set of commands between $\mathcal{M}$ and TS server machine. Our proposed protocol has undergone simulation between role events to facilitate communication among entities and verify security requirements. The events tested include Nisynch, Secret, Commitment, Niagree, and Alive. By utilizing the Scyther tool's send() and rec() directives, we can assess e-commerce requests and identify potential attacks or breaches resulting from the protocol's design. The results demonstrate that our protocol fulfills the requirements for confidentiality (Secret) and transaction efficacy (Alive), ensuring the privacy and availability of information for all parties involved. Commitment: is a specific data agreement, for example, in our proposed protocol, $\mathcal{M}$ was agreed with TS on a combination of nonce ($Salt_i$) and R_i. Niagree: A non-injectable guarantee of agreement is achieved by the proposed protocol. By doing so, the parties' message's integrity can be ensured. Nisynch: The proposed protocol achieves a non-injection synchronization guarantee to ensure that the protocol is against attack.

6.2.2. Scyther Test Results

Here, we present our e-commerce protocol test suggested by the Scyther tool. Figure 6 depicts the test results of our protocol based on the events 'Alive', 'Niagree', 'Nisynch', 'Secret' and 'Commit'. The test displays that public keys (TSKu, MKu), private keys (TSKr, MKr), $\mathcal{M}$ requests, and TS requests are secret. It illustrates that orders are securely exchanged between network entities ($\mathcal{M}$ and TS) without any threats or attacks targeting network entities, security parameters, e-commerce orders sent, and $\mathcal{M}$ data over the network. Our proposed protocol resists attacks in our field of research topic.



Claim				Status		Comments
ElGamal_fernet	TS	ElGamal_fernet,TS1	Secret TSPW	Ok	Verified	No attacks.
		ElGamal_fernet,TS2	Secret XOR(TSPW,XOR(TSPW,TSKr))	Ok	Verified	No attacks.
		ElGamal_fernet,TS3	Secret KeysDivision(k1,k2,k3,k4)	Ok	Verified	No attacks.
		ElGamal_fernet,TS4	Secret XOR(XOR(k1,k2),XOR(k3,k4))	Ok	Verified	No attacks.
		ElGamal_fernet,TS5	Secret MKr	Ok	Verified	No attacks.
		ElGamal_fernet,TS6	Niagree	Ok	Verified	No attacks.
		ElGamal_fernet,TS7	Nisynch	Ok	Verified	No attacks.
		ElGamal_fernet,TS8	Alive	Ok	Verified	No attacks.
		ElGamal_fernet,TS9	Commit M,Er(Er(TSR,XOR(XOR(k1,k2),XOR(k3,k4))),Sal...	Ok	Verified	No attacks.
	M	ElGamal_fernet,M1	Secret MPW	Ok	Verified	No attacks.
		ElGamal_fernet,M2	Secret XOR(MPW,XOR(MPW,MKr))	Ok	Verified	No attacks.
		ElGamal_fernet,M3	Secret KeysDivision(k1,k2,k3,k4)	Ok	Verified	No attacks.
		ElGamal_fernet,M4	Secret XOR(XOR(XOR(XOR(XOR(XOR(k1,k2),XOR(k3,k4)),...	Ok	Verified	No attacks.
		ElGamal_fernet,M5	Secret XOR(TSPW,XOR(TSPW,TSKr))	Ok	Verified	No attacks.
		ElGamal_fernet,M7	Niagree	Ok	Verified	No attacks.
		ElGamal_fernet,M6	Nisynch	Ok	Verified	No attacks.
		ElGamal_fernet,M8	Alive	Ok	Verified	No attacks.
Done.		ElGamal_fernet,M9	Commit M,Concat(Er,Salti),t	Ok	Verified	No attacks.

Figure 6. Validation of the proposed security protocol using the Scyther tool.

6.3. Performance Results and Discussion

The proposed protocol's encryption algorithms were implemented using Java on an Ubuntu 18.04.6 LTS system. The computer used for the study is equipped with an Intel(R) Core(TM) i5-2540M CPU and has 4.00 MB of RAM. To assess the execution time of encoding and decoding in the proposed algorithm, tests were conducted 100 times in Java, and the results were collected. The numerical data obtained were then analyzed in Libre Calc on Ubuntu to generate performance figures and graphs, which will be discussed later. Figure 7 illustrates the original FER algorithm's encryption and decryption time, showing that the decryption process is quicker and has a shorter execution time compared to the encryption process.

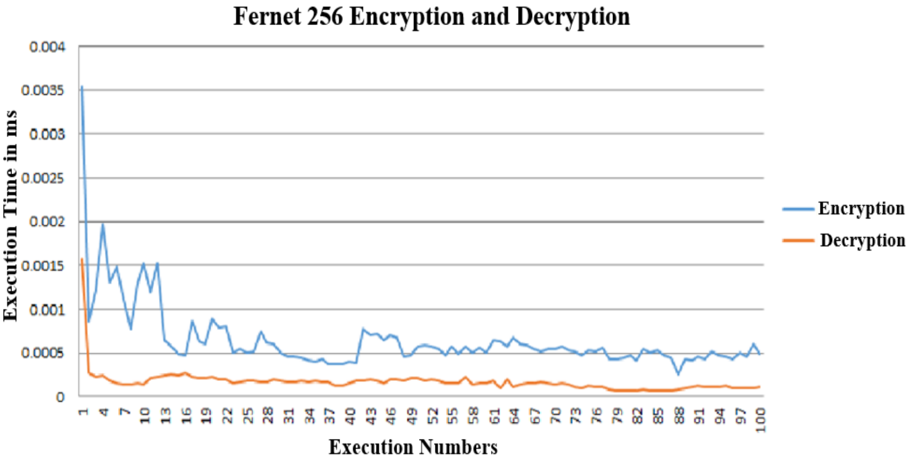


Figure 7. Time required for encryption and decryption for the original Fernet 256-bit.

Figure 8 presents the encryption and decryption speed of the proposed protocol, which combines the ELG and FER algorithms. Each operation in the algorithm, such as text encryption and decryption, iterates 100 times. The slight variation in the speed of encryption and decryption is influenced by the data size each time. The results indicate that the encryption process requires more time compared to decryption. Nevertheless, the overall performance of the encryption and decryption operations remains efficient, with the highest execution time for encryption and decryption being only 0.025 ms.

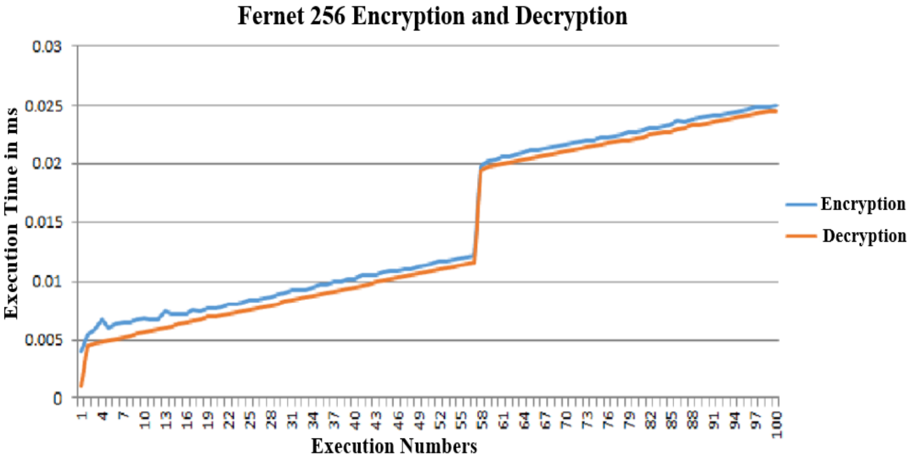


Figure 8. Execution times for encryption and decryption using the proposed algorithm.

Figures 9 and 10 show the performance of purchasing and payment orders by performing the orders' encryption and decryption processes in the proposed approach.

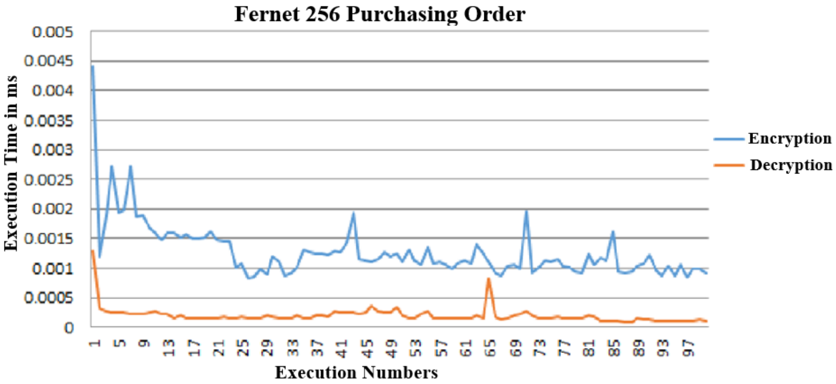


Figure 9. Execution time for the purchasing request encryption and decryption.

Figure 11 depicts the performance of the DLD technique in the proposed protocol through the execution time process of encryption, decryption and key generation. In our proposed protocol, we use DLD technology to detect data leakage and protect the information; however, a data breach or attack cannot be ruled out. We utilize the ELG algorithm to generate a key with a size of 1024, although keys of different sizes and lengths, such as 2048, can also be used. Orders information is exchanged between the $\mathcal{M}$ and the TS in a single network; it can be redeemed for a group of service servers across a range of networks.

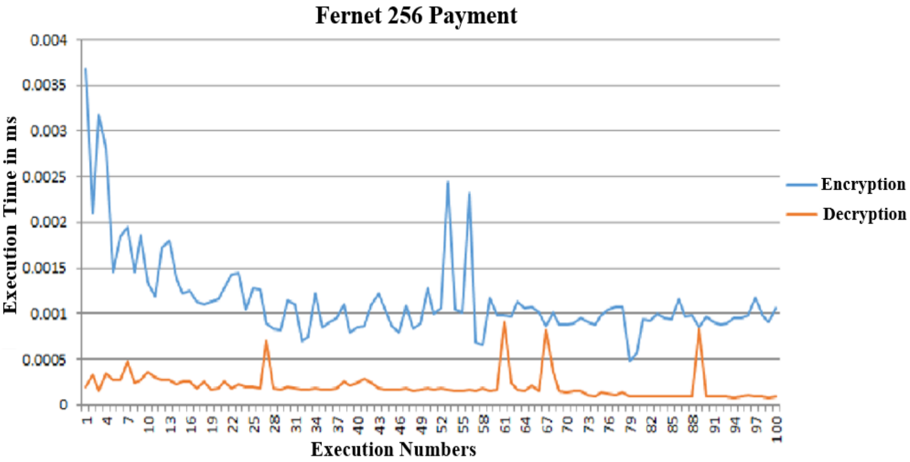


Figure 10. Execution time for the payment request encryption and decryption.

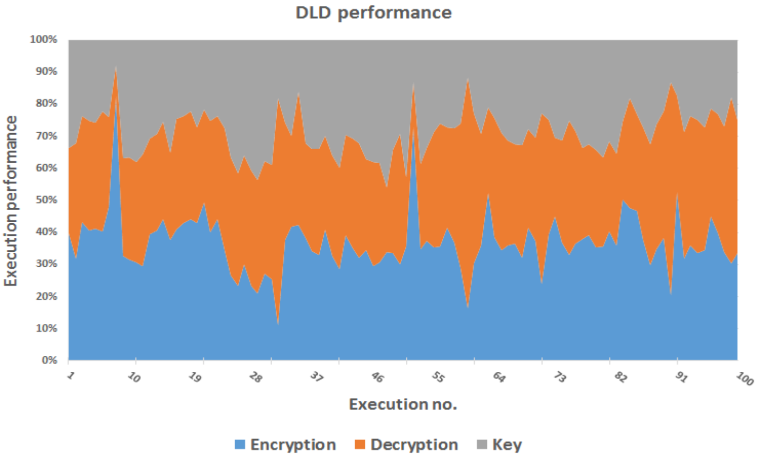


Figure 11. Evaluation of DLD performance depending on the execution time for encryption and decryption and key generation.

In Table 2, we present a comparison of execution times between our proposed protocol and existing protocols. Although there are some variations in algorithms, parameters, and environments, this comparison demonstrates the fairness of our protocol's implementation. As shown in Table 2, our protocol achieves the best execution time among existing protocols for both encryption and decryption operations. When compared to Awan et al. [42], our protocol outperforms theirs, even though they modified the AES algorithm to enhance cryptographic processing speed. However, their protocol lacks the scalability that our protocol, utilizing ELG, offers. Furthermore, Al-gohany and Almotairi [43] employed DES for encryption, but DES is considered compromised based on recent research and is not suitable for safeguarding sensitive information like merchant data. On the other hand, Koppaka and Lakshmi [15] used ELG/AES, Devassy [44] used RSA/AES, and Sylfania et al. [45] used RSA/Blowfish. They incorporated both symmetric and asymmetric cryptography algorithms into their encryption and decryption processes, resulting in increased complexity. In contrast, our protocol relies solely on ELG for generating random keys and FER for performing encryption operations, making it lightweight and efficient for e-commerce applications.

Table 2. Comparison between the proposed cryptosystem, AES, Elgamal+ AES, DES, RSA+AES and RSA+ Blowfish in terms of execution time for encryption and decryption procedures.

Cryptosystem	Encryption Execution Time in ms	Decryption Execution Time in ms
AES [42]	0.1190	0.1481
Elgamal + AES [15]	0.08	1.786
DES [43]	0.062	0.024
RSA + AES [44]	1393	1393
RSA + Blowfish [45]	76.923	84.7826
Proposed	0.00396	0.00101

7. Conclusions

In e-commerce, transaction security is very important, and online transaction security is a key task when it comes to deciding whether to buy a service or product online to safeguard $\mathcal{M}$s' information, as transaction information is not impervious to online fraud. In response to these challenges, we introduced a protocol that ensures robust security for e-commerce transactions through a hybrid encryption approach. Our protocol combines the use of asymmetric keys like ELG keys and symmetric encryption using the FER algorithm, ensuring robust data protection. Additionally, we integrated DLD technology to safeguard information from potential leaks and unauthorized access. To assess the efficacy of our protocol, we subjected it to various e-commerce threats and conducted hands-on testing using the Scyther tool.

As a result, our protocol demonstrates robust resistance against these attacks, effectively ensuring high-performance security against malicious threats. Notably, it achieved the most efficient execution time for encryption (0.00396 ms) and decryption (0.00101 ms) compared to previous studies' algorithms (as shown in Table 2). For future work, we aim to enhance security by incorporating a symmetric random function generator (SRFG) within the FER algorithm to introduce more encryption randomness. Additionally, we plan to subject the protocol to testing against various attacks, including vampire repeat registered attacks. Finally, we intend to implement public key encryption with an equality test (PKEET) in the ELG algorithm, enabling verification without decryption to determine if two encryptions produced from different public keys contain the same e-commerce request.

Author Contributions: All authors provided contributions to the work. Conceptualization, M.A.-Z. and G.S.S.; methodology, M.A.-Z. and G.S.S.; software, M.A.-Z. and G.S.S.; validation, M.A.-Z. and G.S.S.; formal analysis, M.A.-Z. and G.S.S.; investigation, M.A.-Z. and G.S.S.; writing—original draft preparation, M.A.-Z. and G.S.S.; writing—review and editing, M.A.-Z.; supervision, M.A.-Z.; project administration, M.A.-Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable, the study does not report any data.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

In this manuscript, the following abbreviations are employed:

DLD	Data leakage detection
E-commerce	Electronic commerce
ELG	ElGamal algorithm
FER	Fernet algorithm
GM_i	Group of merchants' agents
Hybrid Encryption	Integration of ELG and FER algorithms
$\mathcal{M}$	Merchant
P_i	Leakage probability
Salt	Random value for encryptions and keys
Sco	Score of computation process
D_i	Data group
TS	Trust server

References

1. Kumbhakar, D.; Sanyal, K.; Karforma, S. An optimal and efficient data security technique through crypto-stegano for e-commerce. *Multimed. Tools Appl.* **2023**, *82*, 21005–21018. [CrossRef]
2. Cybersource. 2023 Global Ecommerce Payments and Fraud Report. Technical Report, Cybersource A Visa Solution. 2023. Available online: https://www.cybersource.com/en-us/solutions/fraud-and-risk-management/fraud-report.html#cw-243544106 (accessed on 1 June 2023).
3. RecordedFuture. Annual Payment Fraud Intelligence. Technical Report, Recorded Future Products. 2022. Available online: https://www.recordedfuture.com/annual-payment-fraud-intelligence-report-2022 (accessed on 3 June 2023).
4. Al-Zubaidie, M.; Zhang, Z.; Zhang, J. Efficient and secure ECDSA algorithm and its applications: A survey. *Int. J. Commun. Netw. Inf. Secur. (IJCNIS)* **2019**, *11*, 7–35. [CrossRef]
5. Al-Zubaidie, M.; Zhang, Z.; Zhang, J. REISCH: Incorporating lightweight and reliable algorithms into healthcare applications of WSNs. *Appl. Sci.* **2020**, *10*, 2007. [CrossRef]
6. Al-Zubaidie, M.; Zhang, Z.; Zhang, J. PAX: Using pseudonymization and anonymization to protect patients' identities and data in the healthcare system. *Int. J. Environ. Res. Public Health* **2019**, *16*, 1490. [CrossRef] [PubMed]
7. Jintcharadze, E.; Iavich, M. Hybrid implementation of Twofish, AES, ElGamal and RSA cryptosystems. In Proceedings of the 2020 IEEE East-West Design & Test Symposium (EWDTS), Varna, Bulgaria, 4–7 September 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–5.
8. Al-Zubaidie, M.; Zhang, Z.; Zhang, J. RAMHU: A new robust lightweight scheme for mutual users authentication in healthcare applications. *Secur. Commun. Netw.* **2019**, *2019*, 3263902. [CrossRef]
9. Sidik, A.P.; Efendi, S.; Suherman, S. Improving one-time pad algorithm on Shamir's three-pass protocol scheme by using RSA and ElGamal algorithms. In *Proceedings of the Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2019; Volume 1235, p. 012007.
10. Ali, G.; Dida, M.A.; Elikana Sam, A. A Secure and efficient multi-factor authentication algorithm for mobile money applications. *Future Internet* **2021**, *13*, 299. [CrossRef]
11. Tyagi, S. Enhancing security of cloud data through encryption with AES and Fernet algorithm through convolutional-neural-networks (CNN). *Int. J. Comput. Netw. Appl.* **2021**, *8*, 288–299.
12. Dong, Z. Construction of mobile e-commerce platform and analysis of its impact on e-commerce logistics customer satisfaction. *Complexity* **2021**, *2021*, 6636415. [CrossRef]
13. Abdul Hussien, F.T.; Rahma, A.M.S.; Abdul Wahab, H.B. A secure environment using a new lightweight AES encryption algorithm for e-commerce websites. *Secur. Commun. Netw.* **2021**, *2021*, 9961172. [CrossRef]
14. Kota, C. Secure File Storage in Cloud Using Hybrid Cryptography. Available at SSRN 4209511. 2022. Available online: https://ssrn.com/abstract=4209511 (accessed on 25 June 2023).
15. Koppaka, A.K.; Lakshmi, V.N. ElGamal algorithm with hyperchaotic sequence to enhance security of cloud data. *Int. J. Pervasive Comput. Commun.* **2022**. [CrossRef]
16. Charles, V.B.; Surendran, D.; SureshKumar, A. Heart disease data based privacy preservation using enhanced ElGamal and ResNet classifier. *Biomed. Signal Process. Control* **2022**, *71*, 103185. [CrossRef]
17. Ahmed, S.; Ahmed, T. Comparative analysis of cryptographic algorithms in context of communication: A systematic review. *Biomed. Signal Process. Control* **2022**, *12*, 161–173. [CrossRef]

18. Al-Zubaidie, M. Implication of lightweight and robust hash function to support key exchange in health sensor networks. *Symmetry* **2023**, *15*, 152. [CrossRef]
19. Muhajjar, R.A.; Flayh, N.A.; Al-Zubaidie, M. A perfect security key management method for hierarchical wireless sensor networks in medical environments. *Electronics* **2023**, *12*, 1011. [CrossRef]
20. Parvathi, R.; Girish, M.; Sandeep, M.G.; Abhiram, K. Secured blockchain technology for agriculture food supply chain. *J. Pharm. Negat. Results* **2022**, *13*, 357–361. [CrossRef]
21. He, H.; Zhang, B. Strategy analysis of multi-agent governance on the e-commerce platform. *J. Theor. Appl. Electron. Commer. Res.* **2023**, *18*, 1–18. [CrossRef]
22. Li, Z.; Ren, L.; Li, Z.; Chen, J.; Tian, X.; Zhang, Y. Price dispersion, bargaining power, and consumers' online shopping experience in e-commerce: Evidence from online transactions. *Math. Probl. Eng.* **2023**, *2023*, 6638665. [CrossRef]
23. Sugito, P. Sales multiplize through e-commerce training For Batik craftsman in Paiton Probolinggo. *Empower. Soc.* **2023**, *6*, 9–16.
24. Shyaa, G.S.; Al-Zubaidie, M. Utilizing trusted lightweight ciphers to support electronic-commerce transaction cryptography. *Appl. Sci.* **2023**, *13*, 7085. [CrossRef]
25. Asri, R.; Nasution, M.K.; Suherman, S. Modification of chipertext ElGamal algorithm using split merge. In *Proceedings of the Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2019; Volume 1235, p. 012054.
26. Arboleda, E.R. Secure and fast chaotic ElGamal cryptosystem. *Int. J. Eng. Adv. Technol* **2019**, *8*, 1693–1699.
27. Jain, A.; De, P. Enhancing database security for facial recognition using Fernet encryption approach. In Proceedings of the 2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2–4 December 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 748–753.
28. Prashanth, C.; Teja, D.B.S.; Lavanya, V. *Securing the Data in Cloud Using Fernet Technique*; Technical Report; EasyChair: Stockport, UK, 2022.
29. Habibu, T.; Luhanga, E.T.; Sam, A.E. Developing an algorithm for securing the biometric data template in the database. *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*, 361–371. [CrossRef]
30. Singh, A.; Ikuesan, R.A.; Venter, H. Secure storage Model for digital forensic readiness. *IEEE Access* **2022**, *10*, 19469–19480. [CrossRef]
31. Gupta, I.; Singh, A.K. A holistic view on data protection for sharing, communicating, and computing environments: Taxonomy and future directions. *arXiv* **2022**, arXiv:2202.11965.
32. Patil, R.C.; Kumar, A.; Narmadha, T.; Suganthi, M.; Rao, A.V.S.R.; Rajesh, A. Data leakage detection in cloud computing environment using classification based on deep learning architectures. *Int. J. Intell. Syst. Appl. Eng.* **2022**, *10*, 281–285.
33. WTO. 38 Free Payment Receipt Templates (Excel | Word | PDF). 2023. Available online: https://www.wordtemplatesonline.net/payment-receipt-templates/ (accessed on 20 May 2023).
34. Odunze, D. Cyber victimization by hackers: A criminological analysis. *Public Policy Adm. Res.* **2018**, *8*, 08–15.
35. Kaushik, D.; Gupta, A.; Gupta, S. E-commerce security challenges: A review. In Proceedings of the International Conference on Innovative Computing & Communications (ICICC), Delhi, India, 20–22 February 2020; pp. 1–4.
36. Badotra, S.; Sundas, A. A systematic review on security of e-commerce systems. *Int. J. Appl. Sci. Eng.* **2021**, *18*, 1–19.
37. Alqassab, A.; Hikmat Ismael, Y. EMV electronic payment system and its attacks: A review. *AL-Rafidain J. Comput. Sci. Math.* **2022**, *16*, 23–29. [CrossRef]
38. Xiao, Y.; Zhou, C.; Guo, X.; Song, Y.; Chen, C. A novel decentralized e-commerce transaction system based on blockchain. *Appl. Sci.* **2022**, *12*, 5770. [CrossRef]
39. Liu, X.; Ahmad, S.F.; Anser, M.K.; Ke, J.; Irshad, M.; Ul-Haq, J.; Abbas, S. Cyber security threats: A never-ending challenge for e-commerce. *Front. Psychol.* **2022**, *13*, 4863. [CrossRef] [PubMed]
40. Roy, S.; Sharmin, N.; Acosta, J.C.; Kiekintveld, C.; Laszka, A. Survey and taxonomy of adversarial reconnaissance techniques. *ACM Comput. Surv.* **2022**, *55*, 1–38. [CrossRef]
41. Weichbroth, P.; Wereszko, K.; Anacka, H.; Kowal, J. Security of cryptocurrencies: A view on the state-of-the-art research and current developments. *Sensors* **2023**, *23*, 3155. [CrossRef] [PubMed]
42. Awan, I.A.; Shiraz, M.; Hashmi, M.U.; Shaheen, Q.; Akhtar, R.; Ditta, A. Secure framework enhancing AES algorithm in cloud computing. *Secur. Commun. Netw.* **2020**, *2020*, 1–16. [CrossRef]
43. Al-gohany, N.A.; Almotairi, S. Comparative study of database security in cloud computing using AES and DES encryption algorithms. *J. Inf. Secur. Cybercrimes Res.* **2019**, *2*, 102–109. [CrossRef]
44. Devassy, N. Research Project Questions. Ph.D. Thesis, National College of Ireland, Dublin, Ireland, 2023.
45. Sylfania, D.Y.; Juniawan, F.P.; Pradana, H.A. Blowfish–RSA comparison analysis of the encrypt decrypt process in android-based email application. In Proceedings of the Sriwijaya International Conference on Information Technology and Its Applications (SICONIAN 2019), Palembang, Indonesia, 16 November 2019; Atlantis Press: Amsterdam, The Netherlands, 2020; pp. 113–119.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.