ACTA INFOLOGICA 2023;7(1):1-16
dergipark.org.tr/acin

DOI: 10.26650/acin.1134979

RESEARCH ARTICLE

A Novel Data Encryption Algorithm To Ensure Database Security

Veritabanı Güvenliğini Sağlamak için Yeni Bir Veri Şifreleme Algoritması

Sivan Ibrahim[1], Ahmet Zengin[2], Selman Hızal[3], A. F. M. Suaib Akhter[4], Cevat Altunkaya[5]

[1]Sakarya University, Faculty of Computer and Information Sciences, Department of Computer Engineering, Sakarya, Turkiye

[2](Prof. Dr.), Sakarya University, Faculty of Computer and Information Sciences, Department of Computer Engineering, Sakarya, Turkiye

[3](Assist. Prof.), Sakarya University of Applied Sciences, Department of Computer Engineering, Sakarya, Turkiye

[4](Dr.), Sakarya University of Applied Sciences, Department of Computer Engineering, Sakarya, Turkiye

[5](Dr.), Sakarya University, Faculty of Computer and Information Sciences, Department of Computer Engineering, Sakarya, Turkiye

ORCID: S.S.I. 0000-0003-4707-0578;
A.Z. 0000-0003-0384-4148;
S.H. 0000-0001-6345-0066;
A.F.M.S.A. 0000-0002-2675-1684;
C.A. 0000-0001-5118-2711

Corresponding author:
Selman HIZAL
Sakarya University of Applied Sciences, Department of Computer Engineering, Sakarya, Turkiye
E-mail address: selmanhizal@subu.edu.tr

Submitted: 24.06.2022
Revision Requested: 26.10.2022
Last Revision Received: 10.02.2023
Accepted: 11.02.2023
Published Online: 09.05.2023

Citation: Ibrahim, S.S., Zengin, A., Hizal, S., Suaib Akhter, A.F.M., Altunkaya, C. (2023). A novel data encryption algorithm to ensure database security. *Acta Infologica, 7*(1), 1-16. https://doi.org/10.26650/acin.1134979

ABSTRACT

Many people apply technologies gradually, moving towards new progress and development. Moreover, many of us utilize database systems to assist our work management; the database contains our work or personal information, which increases the risk of losing our data due to disruptive electronic attacks. As a result, protecting databases from electronic attacks and data seizures is crucial. One method of identifying data is through several algorithms so that no benefit is taken from our data during electronic attacks. In this paper, we explain a formula we created for data encryption. The data is encrypted using ASCII Code. Also, we used three keys in the main formula. Because of that formula, each data we save in the database will be encrypted. The data can be Text or numbers. And through using another coordinator with the three previous keys, we can render the data to our original style. The formula focuses on data size and recording speed so that we get the same data size as when the data is encrypted at a reasonable speed.

Keywords: Cryptography, encryption, decryption, cyber, security, database, ASCII code

ÖZ

Günümüzde, özellikle internet üzerinde birçok bulut tabanlı uygulama geliştirilmekte ve insanların kullanımına sunulmaktadır. Bu uygulamaların çoğu veri tabanı sistemlerini kullanır ve özellikle çalışma hayatımızdaki iş veya kişisel bilgilerimizi içerebilir. Kritik bilgilerin bu veri depolarında tutulması yıkıcı elektronik saldırılar nedeniyle veri kaybı riskini artırır. Sonuç olarak, veri tabanlarını elektronik saldırılardan ve veri hırsızlığından korumak oldukça önemlidir ve veri güvenliği sistemlerinin kullanılmasını zorunlu hale getirir. Bu kapsamda çeşitli güvenlik algoritmalarına ihtiyaç vardır. Bu makalede, verilerin şifrelenmesi için yeni bir algoritma önerilmiştir. Üç anahtar ile çalışan algoritmada veriler ASCII Kodu kullanılarak şifrelenir. Çeşitli formüller sayesinde veri tabanına kaydettiğimiz metin ve rakam formundaki her veri şifrelenecektir. Geliştirilen algoritma, verilerin yüksek bir hızda şifrelenmesiyle kayıpsız aynı veriyi tekrar elde etmemizi sağlar. Yapılan testler algoritmanın yüksek veri boyutunda oldukça hızlı çalıştığını ortaya koymuştur.

Anahtar Kelimeler: Kriptografi, şifreleme, şifre çözme, siber, güvenlik, veri tabanı

1. INTRODUCTION

A database is a collection of data that is organized and stored in a structured way so that it can be accessed and modified efficiently. Databases store and manage large amounts of data in a systematic and organized manner. They are an essential component of many modern information systems, allowing organizations to store and retrieve data quickly and accurately (Silberschatz et al., 2002).

Many databases exist, including relational, object-oriented, and NoSQL databases. They can be used for many purposes, such as storing customer information, managing inventory, and tracking financial transactions.

The importance of databases lies in their ability to store and manage large amounts of data in an organized, efficient, and reliable way. They enable organizations to store, retrieve, and manipulate data quickly and accurately, which is essential for various business and scientific applications. Additionally, databases allow for data integration from different sources and enable data sharing and collaboration among other users and systems.

Database security refers to the various measures taken to protect databases from unauthorized access, misuse, or damage. It is an essential aspect of information security that ensures that sensitive data stored in databases is kept safe and secure. Overall, database security is vital to protecting data integrity and confidentiality and helps organizations meet legal and regulatory requirements for data protection (Bertino et al., 2005).

Several articles have been written in the literature. For instance, the authors of (Shelly et al., 2013) concluded that Databases were a preferred target for hackers because of their sensitive and valuable information. A database can be hacked in a variety of ways. A database should be protected from various types of threats and risks. This paper identifies solutions to most attacks, although some are beneficial while others are only momentary. In (Aamer et al., 2005), the writer measured cipher algorithms (AES, DES, 3-DES, and Blowfish) to various data dimensions and encryption periods for two separate devices, a Pentium-4, 2.4 GHz, and a Pentium-II 266 MHz in EBC and CFB Mode. The author decided Blowfish is the quickest, afterward came DES and Triple DES; and CFB needed longer than ECB cipher block mode. Mathur (2012) proposed an encryption algorithm that depended on the ASCII value of the message to be encoded. This algorithm required a key that had the same length as the letter. This key was encrypted and utilized in the encryption and decryption processes. This device required the user to input the key. If the message was longer than the device allowed, the recipient was asked to input a key identical to the range of the letter. This made it difficult for the user to input a large key. Another disadvantage of this algorithm was that it took longer to execute. So, these are two deficiencies of the current algorithm. On the other hand, Nadeem (2006) explored well-known unique vital algorithms, for example, DES, 3DES, AES, and Blowfish. These algorithms were tested, and their efficiency was measured by encrypting input folders of various points and dimensions. Compared to other algorithms, the results revealed that Blowfish performed exceptionally well. It was also discovered that AES outperformed 3DES and DES.

Despite the advancements in database security algorithms, some deficiencies still need to be addressed. Some of the most notable ones include insufficient encryption, lack of access control, lack of data audit trails, and human error. Overall, while current database security algorithms have made significant progress in protecting sensitive data, there is still room for improvement to ensure that databases are more secure and better able to withstand attacks.

This paper's primary goal is to develop a new high-performance and secure data encryption algorithm to overcome the abovementioned issues. In our algorithm, several performance characteristics were considered, such as keys' strength, data size, and the speed of encrypting and decrypting data. In developing the encryption approach, we created the formula composed of four other sub-formulas and three keys so that each is encrypted through those sub-formulas. Then these are all used in the leading formula with an ASCII code to strengthen the protected side of the formula. Finally, we reached the result that the size of the original data and the encrypted data are equal. We showed the developed approach's performance compared with other standard cryptographic algorithms.

The paper's organization is the following: in Section 2, we discussed the important information related to the study. In Section 3, studies related to the proposed system are presented. In Section 4, we gave the proposed system structure with the implementation details. The experimental result and achievements are discussed in Section 5, and finally, the paper is concluded in section 6 with potential future research directions.

2. BACKGROUND

2.1. Database Security

DBSs were created in the 1960s and have become crucial to different firms because this allowed the data to be better coordinated and accessible to consumers. A database (DB) is the collection of data that are connected in pairs and serve in place of information that can be recorded and contain indirect inferences. Databases are categorized based on their architecture: concentrated DB (CDB) and alternatively disseminated DB (DDB). A primary distinction between CDB and DDB is that the CDB saves whole statistics and information in a mere place; however, the DDB protects various parts of DB inside many corporeal places (Emad et al., 2017).

In CDB interference in a site leaves the complete structure inaccessible for any consumer; however, with DDB consumers can visit other DB locations. DBS can be built using different data structures, including comparative patterns, categorized patterns, and patterns toward the object. These DB patterns contain lots of personal information, like credit card use and history, medical records, and pupil history that must be kept secure against unauthorized use. As the risks for DBS were revealed, the necessity for maintaining data security coupled with confidentiality has arisen as a needed safeguard against any threats.

Many approaches have appeared to secure data, and three requirements must be fulfilled to achieve this goal. They are the following: confidentiality, integrity, and availability. Confidentiality is using rules to prohibit an unlicensed person from accessing records. The term “integrity” refers to the assurance that the data is not subjected to any alteration or degradation (Emad et al., 2017). Availability ensures that the customer has consistent and timely access to the information.

The lack of any of these requirements endangers the database. Since the mid-1970s, DB system security has gained a great deal of interest, beginning with entry restriction patterns to DB order that is regarded as one of the early safety approaches for DB safety. Entry restriction is a system that checks a consumer’s privileges in the face of a catalog of authorization to ensure data integrity and confidentiality. Authorization defines which database operations a consumer can carry out and what data the consumer can enter. One more tool for verifying the consumer’s persona is verification which is the first step in accessing the database (Emad et al., 2017).

Also, after verification of the consumer within the system, the database management system (DBMS) contained several mechanisms, for example, inspection inquiry coupled with a display, which kept the data safe from unauthorized interactions. An inspection inquiry is records of activities performed by a particular person in the database. If an illegal procedure is recorded, the database administrator (DBA) investigated the account number that carried it out. A sight approach is a digital table generated by performing comparative exercises on the base table (Kaur et al., 2014).

This approach allowed a user to access a portion of the data while the user cannot directly reach the database. Data secrecy may be protected by the encryption methods used in the figure. Encryption data is done by applying a cipher which makes it indecipherable for other consumers apart from the person having the security clearance to decode the data (Basharat et al., 2012).

2.2. Threats To Database Security

Risks are either problems or activities which may negatively impact DB protection, and they can be deliberate or unintentional (Kaur et al., 2014). The following are some of the most widespread threats to database security:

1. Privilege Abuse: We have two types of Privilege Abuse: Excessive Privilege Abuse (EPA) and Legitimate Privilege Abuse (LPA). EPA occurs when consumers illegally get control of entry rights for a database, which is above their task responsibilities; this excessive privilege could be abused for harmful purposes (Rohilla et al., 2013). The permitted user’s use of legal DB privilege for destructive purposes is called LPA (Rohilla et al., 2013).

2. Privilege Elevation: If the database has a loophole, an attacker might be able to manipulate it to change the entry requirements for all consumers from regular customers to company management (Rohilla et al., 2013).
3. SQL Injection: SQL Injection occurs when an assailant enters sequences of illegal SQL statements into a vulnerable SQL data means. By applying SQL injection, assailants can receive full entry for the whole database (Singh et al., 2014).
4. Platform Vulnerabilities: Vulnerabilities in performing systems and whatever external assistance is enabled on a DB server may cause DB harm, for instance, disallowed entry, rejection of service, and, ultimately, data exploitation (Singh et al., 2014).
5. Weak Audit Trail: Audit trails are designed to save a consumer's actions inside the database. As a result, the lack of an audit trail endangers the organization's databases (Basharat et al., 2012).
6. Rejection of Assistance: This threat inhibits permitted consumers from entering the DB. It threatens every company (Basharat et al., 2012).
7. Weak Authentication: Weak verification can enable hackers to employ techniques, for example (community engineering and violent strength) for hacking legitimate consumers' usernames and passwords to subsequently enter the database (Singh et al., 2014).
8. Backup Data Exposure: Since backup DB storage media is rarely secured from any threat, most often safety violations involve stealing hard disks as well as backup tapes (Singh et al., 2014).

2.3. Database Security Measures

Risks are either problems or activities which may negatively impact DB protection, and they can be deliberate or unintentional (Kaur et al., 2014). The following are some of the most widespread threats to database security:

Security methods that protect the database against unauthorized individuals, intentional attacks, data leakage, and hackers (Kaur et al., 2014). It covers many problems, including lawful, moral, regulation, and order-connected concerns. Database safety is a complex step that every company should strengthen to perform its actions smoothly and effectively. Any good company demands that the security and privacy of its database be protected from unauthorized entry, malicious intent, and unintended alteration (Basharat et al., 2012).

Data security is achieved through various sides of a data control scheme (DBMS). DBMS is a collection of implementations that handle the data stored inside a database and aids in data organization for better performance (Kulkarni et al., 2012).

To minimize risks, all DBMSs have security strategies developed for these purposes (Patil et al., 2012). Many security mechanisms have been developed to secure databases.

The four main security mechanisms that are applied to protect DBs from attacks are as follows: entry restriction, disruption management, flow management, and figure encryption.

Figure 1 depicts these monitoring steps. As previously said, entry management is a procedure that ensures data security by comparing the user's privileges to a list of licenses.

The mentioned licenses are managed using a Discretionary Access Control (DAC) approach, a Mandatory Access Control (MAC) approach, a Role-Based Access Control (RBAC) approach, or an Attribute-Based Access Control (ABAC) approach (ABAC). When individuals are allowed entry to analytical or synopsis data, the access control system prohibits them from gathering sensitive information that is not part of their user privileges.

Flow control guarantees that no unauthorized users can access the records. Encryption (as previously stated) is considered an act of transforming data by applying a cipher to cause it to be illegible for all consumers but the person having a password and user privilege to decode the figure (Kaur et al., 2014).

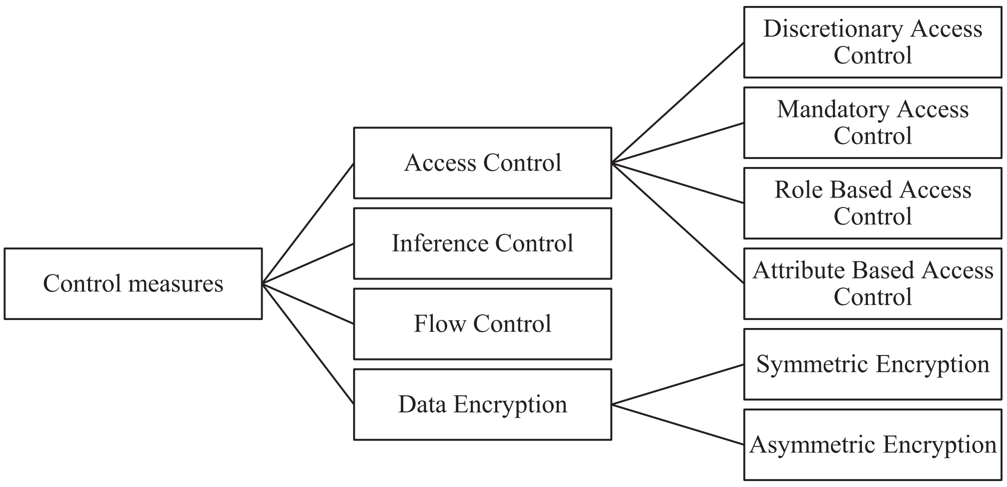


Figure 1. Control measures to protect DBs

2.4. Data Encryption

As data is encrypted with a cipher, it becomes unreadable to most consumers apart from people having the proper password to decode it (Basharat et al., 2012). Although hackers could compromise entry restriction procedures, encryption entries are still required for decoding the data (Thuraisingham et al., 2015).

Encryption stages rely on the algorithm and the entrance applied to encrypt the figure. There are two kinds of encryption: ordered encryption as well as deformed encryption. Encryption can be performed at three different stages, as seen below. Fig. 2 depicts them:

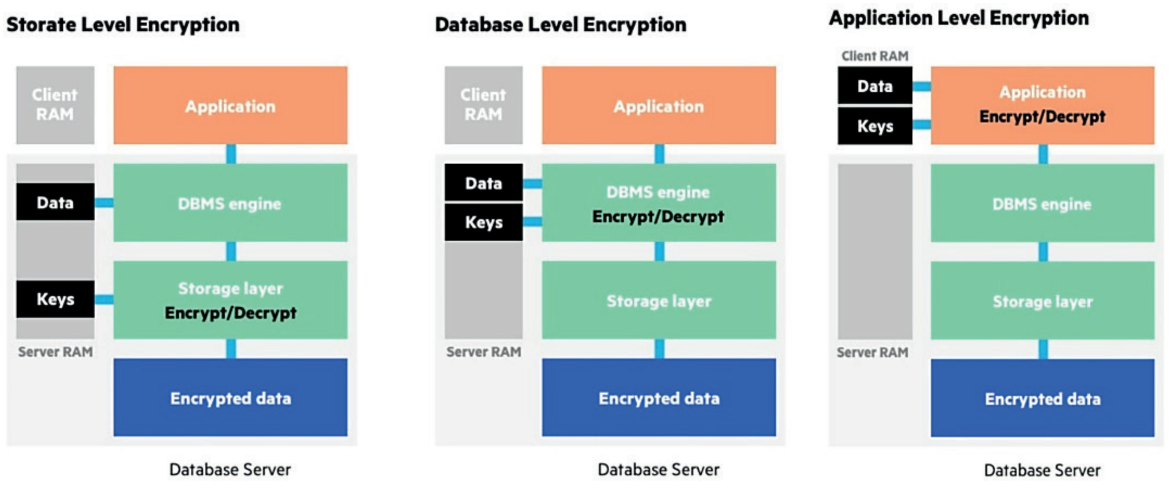


Figure 2. Levels of encryption

Storage-Level Encryption: storage-level encryption encrypts data inside the storehouse component, protecting it, for instance, against television stealing). It is ideal for encrypting folders, and whole folders within a performing order circumstance. Storage-level encryption has the benefit of being transparent from a database standpoint, preventing any updates to the current application. However, as the storehouse component is unaware of objects of the database as well as order, the encryption technique is not likely to be linked to consumer rights, for example, applying different encryption entries for different consumers) or data privacy.

As a result, choosy encryption – that is, encrypting database parts for reducing encryption elevated –is restricted to the folder granularity. Furthermore, discriminatingly encrypting folders is unsafe as no copy of confidential data can be left unencrypted, for instance, in log folders, momentary files, and so on. (Luc et al., 2010).

Database Encryption (DLE): database-level encryption secures data since it is being put into alternatively extracted relative to a database. Thus, the encryption technique could get integrated into database structures and linked to data privacy and consumer rights. Using discriminating encryption is feasible and could be conducted at different levels of granularity, including rows, columns, and tables. It could also get linked to logical requirements (for example, encrypt wages greater than 10K€ per month).

For both methods, data is decrypted on the database server at duration. In this way, on the server side, the encryption entries must be transferred and held, coupled with encrypted files, offering a poor level of security to stop an unprincipled server administrator or an attacker impersonating the administrator. Hackers could also follow the history and find encryption keys or plain text data (Luc et al., 2010).

Application-Level Encryption: Application-level encryption transfers the encryption/decryption mechanism toward the data-generating devices. Encryption is conducted in the program, bringing the data to the network. The data is transmitted encrypted, customarily kept, and recovered encrypted, to be eventually decrypted inside the application.

Since the keys should never depart the application side, this technique has the advantage of separating encryption keys from the encrypted data contained in the database. But, to implement this approach, programs must be modified. Furthermore, relying on the encryption granularity, the program could be required to retrieve a significant collection of data than that allowed to the user, thereby creating a security leak. Indeed, the user (or other hacker obtaining entry to the computer that the program works) can access the program and receive illegal entry to data. Finally, such a technique results in output overheads (indexes on encrypted information are purposeless) and also prevents the employment of progressive database workability like saved formulas (i.e., code saved within the DBMS that could be exchanged and used via multiple programs) as well as triggers on encrypted data (i.e., code fired while information within the database is added).

Application-level encryption has the most flexibility regarding granularity and essential control since the encryption granularity, and encryption keys can be selected based on application logic (Luc et al., 2010).

2.5. ASCII Code

American Standard Code for Information Interchange (ASCII) is a symbol-representation code that applies data. Each letter is a designated number between 0 and 127. Separate data is designated for each capital as well as a little letter.

As seen in the ASCII below, the capital letter A was designated the digit 65. However, the little letter was assigned the decimal number 97. ASCII code returns to the teletype's times and mechanic-like printers, because it precedes the Internet. Management codes for ASCII decimal data are limited to numbers between 0 to 31 and are not widely available anymore.

However, if you want to attempt to play alongside association procedures, you can view the restriction codes in the application. The restriction codes are explained in the ASCII Control Codes table. The entries you click and the correspondences are obtained like a sequence of data while a device delivers the results. Letters that you wrote or made are symbolized through this data. Because ASCII is limited to numbers between 0 and 127, 7 bits or 1 byte of information is required. To give an ASCII example of string coding, it could be 99 97 99 116 117 115 46 105 111. These coding bits, as well as bytes, are understood by microprocessors. All is a series of pieces to it.

2.6. Cryptography

Cryptography is a method of protecting data from unauthorized entry. It is made up of two major parts:

Nowadays, cryptography has turned into a requirement to all firms. Data safety is a critical element in a firm for maintaining and protecting their firm's data from rival firms. It also assists in ensuring a consumer's confidentiality. Currently, codes could be more dependable regarding this duty as it is effortless to find codes because of the limited scope. Furthermore, if the passcode is not strong, it is relatively easy for a hacker to break the it. (Kakkar et al., 2010).

Accordingly, to preserve data, different algorithms have been developed. It assists in carefully preventing the unauthorized use of and maintaining the privacy and security of bank accounts, electronic repositioning of financial supports, and a lot of daily life consequences.

3. PREVIOUS WORKS

As new technology is developed, ensuring the security of databases is an area that has received long term and ongoing research.. Some of the works related to that research work are listed in this section.

Authors in (Diaa et al., 2010) investigated the Symmetric Encryption Algorithms' performance. This study evaluated six exceptional, widespread encryption algorithms: AES (Rijndael), DES, 3DES, RC2, Blowfish, and RC6. A differentiation was implemented at different setups per algorithm, like different dimensions of data blocks, various data types, battery power consumption, various key dimensions, and consequently, encryption/decryption quickness. The empirical model indicated the succeeding consequences.

No crucial difference existed in the exhibition of the results, whether in hexadecimal base encoding or foundation 64 encodings. When modifying packet dimension, it was discovered that RC6 demands fewer durations when compared to the total algorithms excluding Blowfish. When it came to changing data categories like images in the position of text, it was discovered that RC2, RC6, and Blowfish contained drawbacks as compared to other algorithms concerning spending time. Furthermore, 3DES, up to now, had weaker operations than algorithm DES. Lastly, when modifying the key dimension (feasible just in AES and RC6 algorithms), it showed that a more significant key dimension caused the evident difference in the battery and time-wasting.

The authors of (Hamdan et al., 2010) conducted a relative dissection of three encryption algorithms (DES, 3DES, as well as AES) based on nine criteria, including key breadth, cipher kind, block dimensions, safety, possible keys, feasible ASCII can be printed letter keys, as well as duration, taken for searching the total feasible keys at 50 billion keys each moment, and so on. According to research, AES was superior to DES and 3DES.

Authors in (Agarwal et al., 2010) provided a thorough review of common symmetric key encryption algorithms such as DES, TRIPLE DES, AES, and Blowfish. Symmetric Key algorithms, such as RSA, were quicker than Asymmetric Key algorithms. Furthermore, Symmetric algorithms needed less memory than Asymmetric encryption algorithms. Moreover, Symmetric-key encryption was more secure in comparison to Asymmetric key encryption.

Authors in (Singh et al., 2013), showed that the AES algorithm was efficient in quickness, duration, quantity, and outpouring impact.

The author in (Mittal, 2012) discovered that the algorithm's capability depended on the key breadth. Key breadth was a straight comparable for safety and conversely a comparable for Implementation. When the key breadth rises, the algorithm's security expanded; but, the operation decreased.

The authors in (Sesay et al., 2005) examined the importance of cryptography in database security. Confidential data stored in the clear in database systems was subject to attack. No matter how many security mechanisms were implemented, there were certain security flaws that attackers could exploit to access the database. However, information leaks could be avoided by encrypting confidential data before storing it in the archive. And the whole database security problem could be reduced to protecting a few cryptographic keys.

This method is based on ASCII values in (Arya et al., 2017). For encryption and decryption, ASCII characters utilized with string length accompanied by numerical calculations. To crack the procedure, the attacker needs a lot of information about the plain text; a single piece of information, such as string length, is insufficient. The use of different string lengths strengthens the technique. Further operations apply and are dependent on the size of the string. Consequently, that technique does not depend on any specific key or key generation method. It is the strength of the method.

Authors in (Shinge et al., 2014) suggested a symmetric encryption algorithm applying ASCII prices of the figure. The algorithm provides positive leads to little performing duration. This method produces a key for encrypting the letter. The spontaneously produced key is changed to a distinct string, and the identical key is applied to encryption and decryption. Consequently, this algorithm is identified as the symmetric-key algorithm.

Authors in (Seth et al., 2011) performed the dissection of three algorithms: RSA, DES, and AES; while considering some parameters, for example, the length of duration, memory consumption as well as output byte. The frameworks are a primary problem to worry about in all Encryption Algorithm. Empirical results indicate that the DES algorithm requires minimum encryption duration and the AES algorithm contains minimum memory application; however, the encryption duration variety is a small problem when it comes to AES and the DES algorithm. The RSA algorithm requires the largest encryption duration and memory utilization, which is extremely big; however, the output byte is minimal.

The authors (Kulkarni et al., 2012) deduced that databases were the primary factor to whatever varieties of implications. The database includes extremely significant and private data so hacks are very likely. Different types of risks on databases were explored in this study. The revision of effective database safety methods, such as entry control, methods relative to SQLIA, encryption, and data deforming, was explored.

The authors of (Elminaam et al., 2009) provided an implementation assessment of various symmetric encryption algorithms. The selected algorithms were AES, DES, 3DES, RC6, Blowfish, and RC2. They discovered that when the packet size was changed, Blowfish outperformed other encryption algorithms.

4. PROPOSED METHOD

The formula is clarified using a working style chart seen in Figure 1. We used these in our formula:

1. 3 keys are created by the user, one by the programmer, and the other in a variable way.
2. Two texts to create key1 and key2.
3. Ascii Code.
4. Four Sub formulas

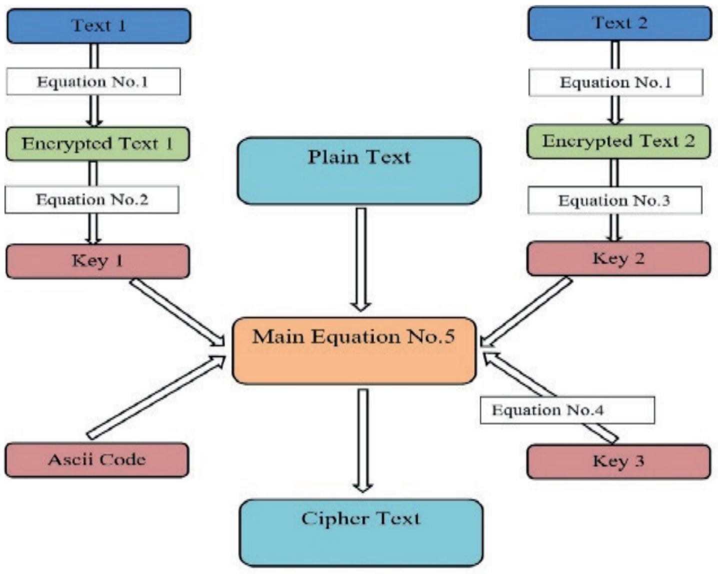


Figure 3. Equation diagram

In general, our formula was applied in the following steps:

In the first step, we needed two texts, one written by the user and the other by the programmer. The number of characters in the text had to be between 8 and 50. The character of the text was one of these characters (a-z, A-Z, 0-9, Keyboard Symbols). We encrypted both texts through formula number 1.

For example, Text1 = Ah5$Z2t*K7 and Text2 = @mQ4s#D7$L

So: Eq (1) = (Ascii code for character + sum of character number in Text)

Table 1
Examples of encrypting text (1) in equation (1)

Character	**A**	**h**	**5**	**$**	**Z**	**2**	**t**	*****	**K**	**7**
Ascii code	65	104	53	36	90	50	116	42	75	55
Eq(1)	65+10	104+10	53+10	36+10	90+10	50+10	116+10	42+10	75+10	55+10
New Ascii code	75	114	63	46	100	60	126	52	85	65
New Character	K	r	?	.	d	<	~	4	U	A

Table 2
Examples of encrypting text (2) in equation (1)

Character	**@**	**M**	**Q**	**4**	**s**	**#**	**D**	**7**	**$**	**L**
Ascii code	64	109	81	52	115	35	68	55	36	76
Eq(1)	64+10	109+10	81+10	52+10	115+10	35+10	68+10	55+10	36+10	76+10
New Ascii code	74	119	91	62	125	45	78	65	46	86
New Character	J	W	[	>	}	-	N	A	.	V

Text1 changed to Kr?.d<~4UA, and Text2 changed to Jw[>}-NA.V

In the second step, we got both Key1 and Key2 by applying two different formulas (formula number 2 on text 1 and formula number 3 on text 2).

Eq (2) = ((total number of all encrypted character ascii code number in Text1) * 2) / (sum of character number in Text1 -2) = Key 1

Table 3
Key (1) generating equation

Character	**K**	**R**	**?**	**.**	**d**	**<**	**~**	**4**	**U**	**A**
Ascii code	75	114	63	46	100	60	126	52	85	65
Eq(2)	((75+114+63+46+100+60+126+52+85+65) * 2) / (10 - 2) = (786 * 2) / 8 = 196.5									

Key1 = 196 Because Key1 should be integer number

Eq (3) = ((total number of all encrypted character ascii code number in Text2) - Key1) / (sum of character number in Text2 -1) = Key 2

Table 4
Key (2) generating equation

Character	**J**	**w**	**[**	**>**	**}**	**-**	**N**	**A**	**.**	**V**
Ascii code	74	119	91	62	125	45	78	65	46	86
Eq (2)	((74+119+91+62+125+45+78+65+46+86) - 196) / (10 - 1) = (791 - 196) / 9 = 66.11									

Key2 = 66, because Key2 should be an integer number like Key1.

In the third step, by applying a sub-formula number 4, we got Key 3, which was a variable key and shifted according to the number of characters of the text we encrypted.

Key3= sum of the numbers in the ASCII code of the letter

In the fourth step, we got our encrypted letter using all three keys (1, 2, 3), with ASCII code for characters in the main formula.

We applied the formula to the plain text (Sakarya) when it was known that Key1 =196 and Key2 =66

Eq (5) = (Ascii code for character + Key1) + (Key2 - sum of character number in Key2) + Key3 = New Ascii code. For Encryption:

Table 5
Encrypting procedure

Plain Character	**Ascii code**	**Key1**	**Key2**	**Key3**	**Encryption Equation: (Ascii code + K1) + (K2 - Length of K2) +K3**	**New Ascii code**	**Cipher Character**
S	83	196	66	11	(83 + 196) + (66 - 10) + 11	346	Ś
A	97	196	66	16	(97 + 196) + (66 - 10) + 16	365	ŭ
K	107	196	66	8	(107 + 196) + (66 - 10) + 8	367	ů
A	97	196	66	16	(97 + 196) + (66 - 10) + 16	365	ŭ
R	114	196	66	6	(114 + 196) + (66 - 10) + 6	372	Ŵ
Y	121	196	66	4	(121 + 196) + (66 - 10) + 4	377	Ź
A	97	196	66	16	(97 + 196) + (66 - 10) + 16	365	ŭ

Eq (5) = (New Ascii code + Key1) - (Key2 - sum of character number in Key2) - Key3 = Ascii code

For Decryption:

Table 6
Decrypting procedure

Cipher Character	**New Ascii code**	**Key1**	**Key2**	**Key3**	**Decryption Equation (New Ascii code + K1) - (K2 - Length of K2) - K3**	**Ascii code**	**Plain Character**
Ś	346	196	66	11	(346 - 196) - (66 - 10) - 11	83	S
ŭ	380	196	66	16	(380 - 196) - (66 - 10) - 16	97	A
ů	372	196	66	8	(372 - 196) - (66 - 10) – 8	107	K
ŭ	379	196	66	16	(379 - 196) - (66 - 10) – 16	97	A
Ŵ	381	196	66	6	(381 - 196) - (66 - 10) – 6	114	R
Ź	401	196	66	4	(401 - 196) - (66 - 10) – 4	121	Y
ŭ	367	196	66	16	(367 - 196) - (66 - 10) – 16	97	A

If the data type is number or currency, then: Eq (5) = (value of (number or currency) / 5) + 5

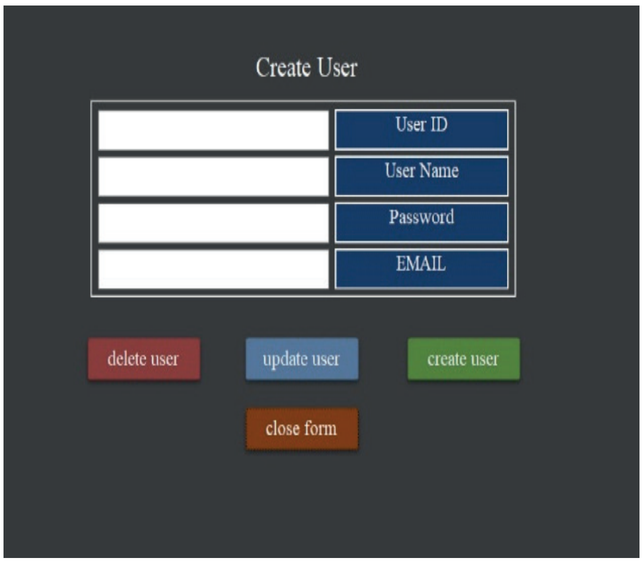
For example: How to encrypt 550, Encrypted value = (550 / 5) + 5 = 110 + 5 = 115

Eq (5) = ((Encrypted value) - 5) * 5

Now, to decrypt 115, decrypted value = (115 – 5) * 5 = 110 * 5 = 550

4.1. Implementation

To test the formula, we designed a database from Microsoft access 2016 that consisted of two parts; one was to insert some data into the database table, and the other was to test our formula's speed. To access our database, a user must be created, as seen in Figure 4. Then, as seen in Figure 5. we logged into the database using the user and password that were created.



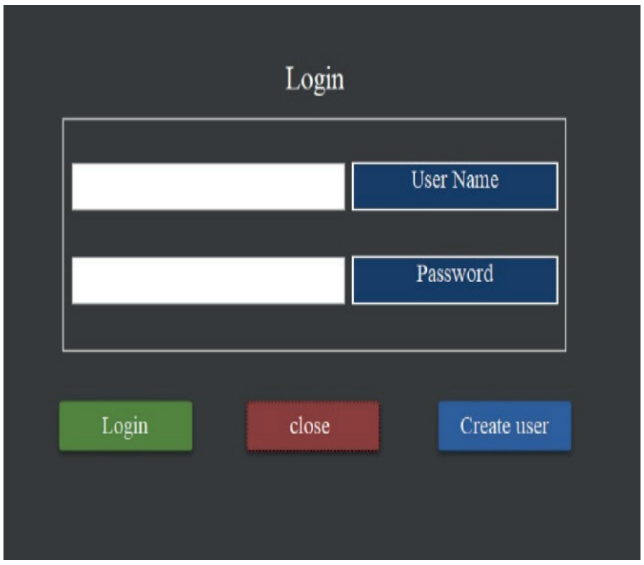
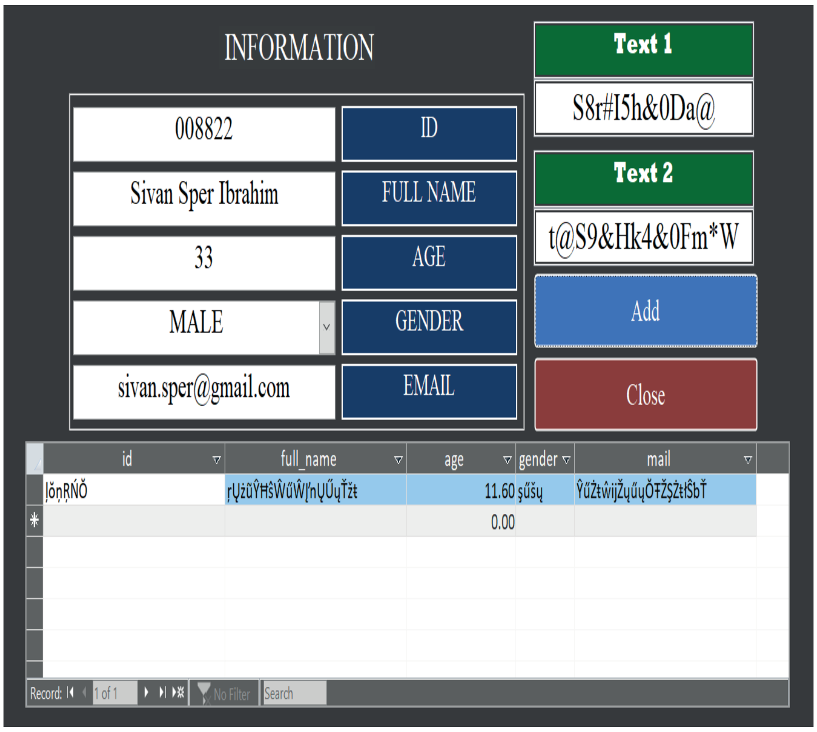


Figure 4. Create User

Figure 5. Login form

Next, we opened a form with two sections, one of which was for data entry and the other for data recording, where Text 1 and Text 2 were used to enter information such as (Id, Full name, Age, Gender, Email) after which we saw that the data was encrypted as seen in Figure 6.

In the second section, we inserted Text 1 and Text 2, and then we inserted our plain text, which, after we pressed the encrypt command, produced ciphertext in one text box and the time to encryption in another text box. For decryption, we followed the same procedure as seen in Figure 7.

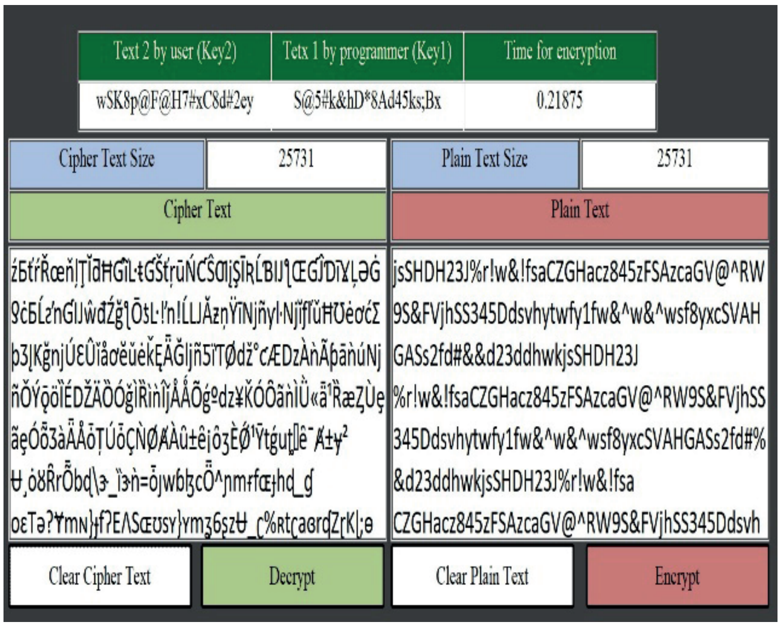


ID	008822
FULL NAME	Sivan Sper Ibrahim
AGE	33
GENDER	MALE
EMAIL	sivan.sper@gmail.com
Text 1	S8r#I5h&0Da@
Text 2	t@S9&Hk4&0Fm*W

id	full_name	age	gender	mail
ļŏņŖŃŎ	ŗŲżũŶĦŝŴűŴĮ'nŲŰųŤžŧ	11.60	şűšų	ŶűŻŧŵĳŽųűųŎŦŽŞŻŧĺŜbŤ
		0.00		

Record: 1 of 1 No Filter Search

Figure 6. Encryption form



Text 2 by user (Key2)	Tetx 1 by programmer (Key1)	Time for encryption
wSK8p@F@H7#xC8d#2ey	S@5#k&hD*8Ad45ks;Bx	0.21875

Cipher Text Size	25731	Plain Text Size	25731

Cipher Text

```
źБtŕŘœňĮŢĬƌĦĜĩL·ŧƓŠťŗūŃƇŜƢĮŞĪŖĹƁIJ¶ŒƓĴƊĩƔĻəĠ
ƍċБĹƨnƓIJŵđŹğ¶ŌƭĿ·!ŉ!ĹLJĂƶņŸïNjñƴŀ·NjïƒĨŭĦƱėσćΣ
þ3ĮĶğnjÚƐÛïåơĕŭėkĘǞĞljñƼïŦØdž°ƈǼDzÀǹÃþāǹúNj
ñŎŶǭöȈĒDŽÄȌÓğȈȐìǹĬĵÅǺÕǵºdz¥ǨÓÔãǹÌǛ«ǟ¹ȐæŽÙę
ãęÓȭ3àǞǺōŢÚōÇǸØǼÀû±êįôʒĖǾ¹Ÿtǵuţ⁅ê¯Ⱥ±Ɏ²
Ʉ¸ȯȣȒrȬbȡ\ɜ_ȉɝǹ=ȭȷwɓƺcȪ^ɲmɟfœɟhȡ_ɠ
oɛTə?ɄmɴɄɟf?EɅSɶʊsʏ}ʏmʒ6ʂzɄ_ɕ%ʀtɕaɞrɖZɼK|;ɵ
```

Plain Text

```
jsSHDH23J%r!w&!fsaCZGHacz845zFSAzcaGV@^RW
9S&FVjhSS345Ddsvhytwfy1fw&^w&^wsf8yxcSVAH
GASs2fd#&&d23ddhwkjsSHDH23J
%r!w&!fsaCZGHacz845zFSAzcaGV@^RW9S&FVjhSS
345Ddsvhytwfy1fw&^w&^wsf8yxcSVAHGASs2fd#%
&d23ddhwkjsSHDH23J%r!w&!fsa
CZGHacz845zFSAzcaGV@^RW9S&FVjhSS345Ddsvh
```

Figure 7. Speed test form

5. RESULTS AND DISCUSSION

This section compared our algorithm to other algorithms (Blowfish, DES, 3DES, RSE, and RSA). For instance, if we have data as shown in Table 7, we reached the results using the following::

Key1 = S8r#I5h&0Da@

Key2 = t@S9&Hk4&0Fm*W

1. When we encrypted plain text, it provided us with the same size as plain text.

2. If a character appeared in the text more than once, it was encrypted with a new character each time.

3. When we encrypted text-type data, it gave us text-type data, but if we encrypted number-type data, we got number-type data, which was another significant point for the algorithm speed when searching for specific data.

4. Another essential point in our algorithm was that when we encrypted different characters in the plain text, the matched character appeared in the ciphertext. This was significant for data security because it would be impossible to convert the encrypted character back to the original if an attacker took over the data.

5. Another point was the length of our Keys, which were between 8 and 50 characters, which was a vital factor in protecting our algorithm because breaking the Keys would require more time and effort.

To clarify the above-mentioned points in our experimental results, we provided two charts for both encrypted and decrypted texts.

Figure 7 represents the original plain text which holds 52 frequencies the same size as the text, while Figure 7. represents the secured text which holds 1039 frequencies. This proves that the proposed method works well on the suggested text.

Table 8
Data encryption in our system

Field name	Id	Full name	Age	Gender
plain text	008821	Sivan Sper Ibrahim	33	MALE
cipher text	ļŏņŖŃō	ŗŲżũŶĦŝŴűŴĮŉnŲŰųŤžŧ	11.6	şűšų

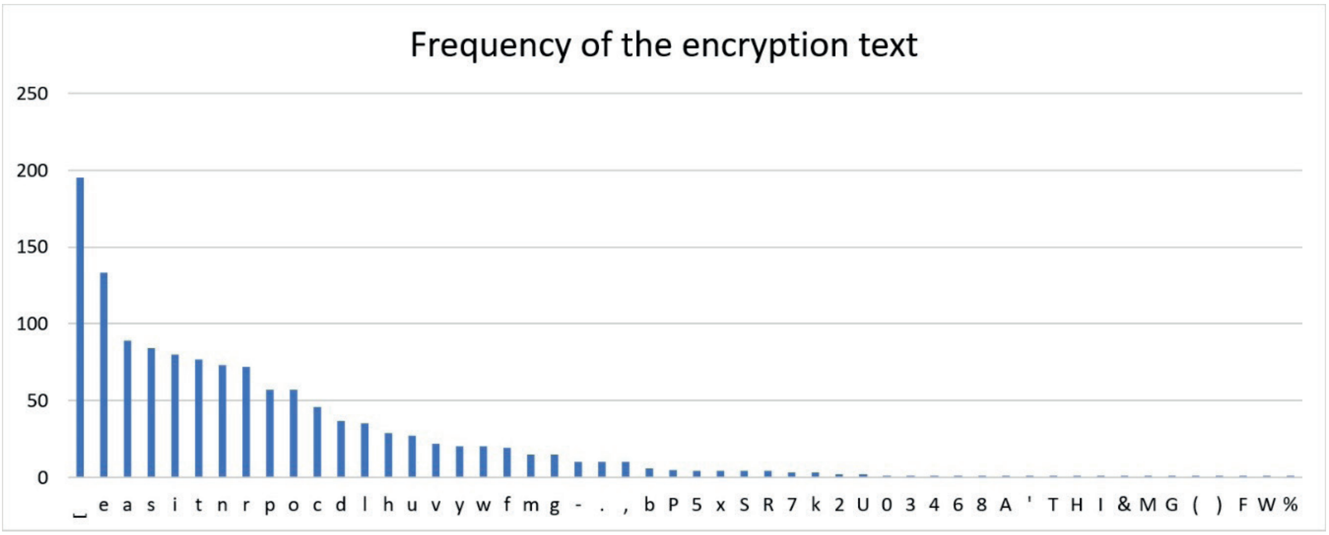


Figure 8. Frequency of the encryption text

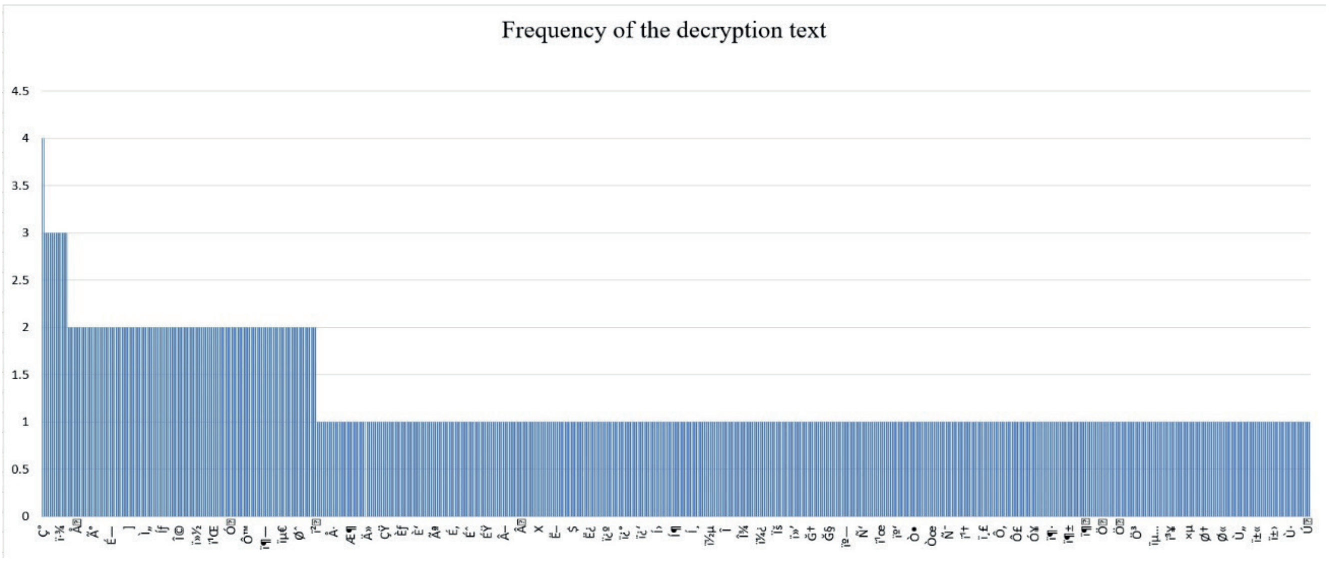


Figure 9. Frequency of the decryption text

To compare the speed of our algorithm, we used the previous research (Patil et al., 2016). The author showed that when a 25KB file (RSA) was encrypted, it took more time than other algorithms, but (Blowfish) took the least amount of time, as shown in Figure 8.

When we conducted similar research on a decrypted 25kb file (RSA), it took more time than other algorithms, but (Blowfish) took the least time, as shown in Figure 9.

To compare algorithm speeds for a 25 KB file, we tested our method. We found that it required 256ms for encryption and 272ms for decryption after five repetitions, indicating that our algorithm was quicker than all of the (RSA, DES, TDES, and RSA) encryption as decryption speeds, as illustrated in Figure 10.

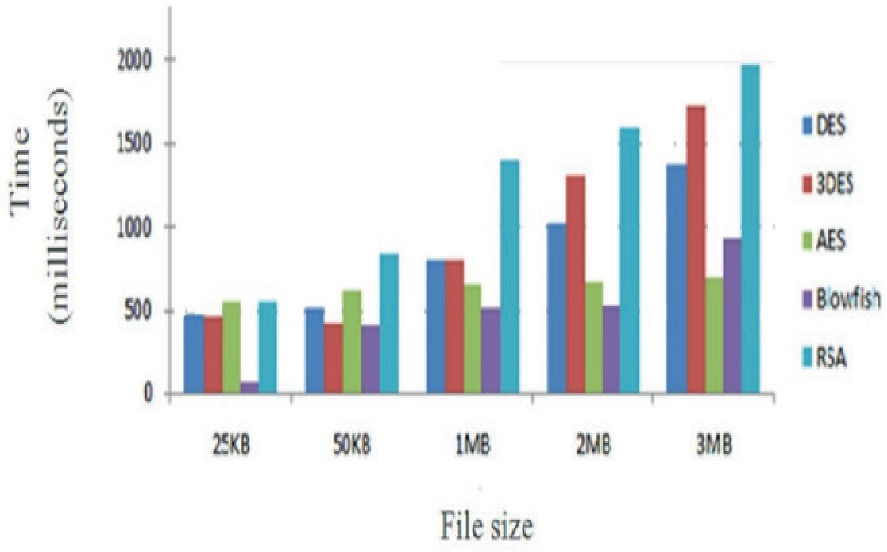


Figure 10. Encryption time vs. file size for DES, 3DES, AES, Blowfish, and RSA (Patil et al., 2016)

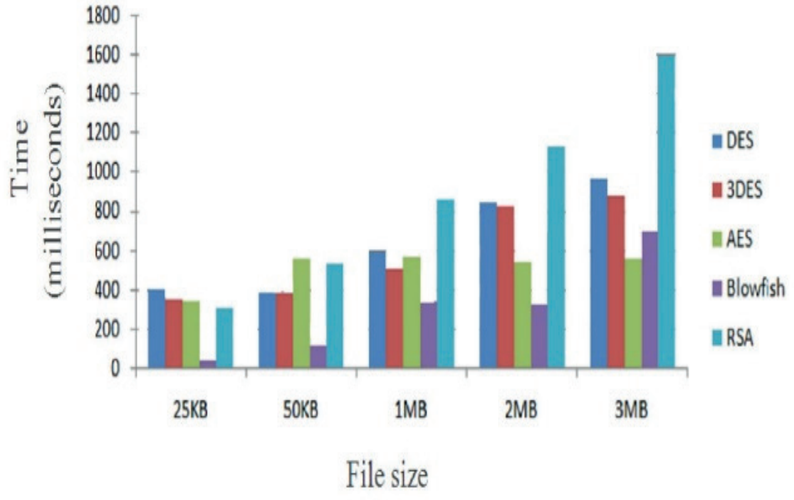


Figure 11. Decryption time vs. file size for DES, 3DES, AES, Blowfish, and RSA (Patil et al., 2016)

To compare the algorithm speeds for a 25 KB file, we tested our method. It required 256ms for encryption and 272ms for decryption after five repetitions, indicating that our algorithm was quicker than all the (RSA, DES, TDES, and RSA) encryption as decryption speeds, as illustrated in Figure 11.

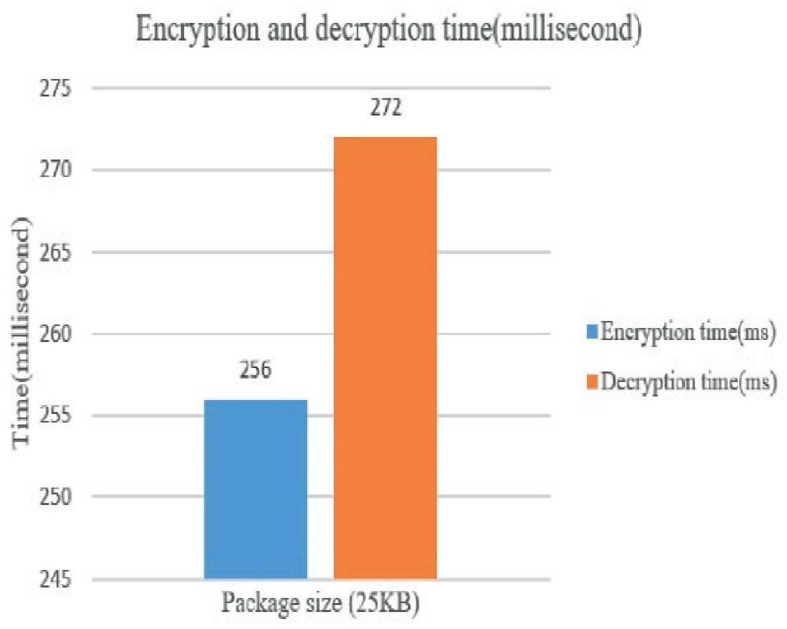


Figure 12. Encryption and decryption time

As a result, we will use the algorithm to create databases and encrypt data because it has large and fast data encryption security, which will secure our data from attacks.

6. CONCLUSIONS AND FUTURE WORK

Cryptographic algorithms protect the confidentiality, integrity, and authenticity of data stored in databases. Confidentiality is the property of ensuring that sensitive information is not disclosed to unauthorized individuals. Cryptographic algorithms can encrypt data stored in a database so that even if an attacker gains access to the database, they will not be able to read the data without the decryption key. Integrity ensures that unauthorized individuals have not modified or tampered with data. Cryptographic algorithms create a message digest of the data, which can then be stored in the database and the data itself. The message digest is a unique representation of the data produced by applying a mathematical function to the data. If the data is modified, the message digest will also be different, allowing for the detection of any changes to the data.

Authenticity ensures that data coming from the source it claims to come from has not been tampered with in transit. Cryptographic algorithms can be used to sign data using a private key so that the recipient can verify the authenticity of the data using the corresponding public key.

Overall, cryptographic algorithms play a critical role in ensuring the security and integrity of data stored in databases. This paper is a comparative study between created algorithms for data encryption used when creating databases against several different cryptographic algorithms, including (DES, 3DES, AES, Blowfish, and RSA) in terms of security and speed of encryption and decryption. The results revealed that the created algorithm was faster than the (DES, 3DES, AES, and RSA) algorithms but slower than the (Blowfish) algorithm. Also, in the created algorithm, the plain text and the ciphertext were the same size. When we encrypted text-type data, we got text-type data. But if we encrypted number type-data, we got number-type data. The developed algorithm is secure. If there is more than one character in the text, each will encrypt with a different character. Also, when we encrypted various characters in plain text, the matched character appeared in the ciphertext. In the future, we will focus on the created algorithm, which can be compared with other cryptographic algorithms.

Peer-review: Externally peer-reviewed.
Author Contributions: Conception/Design of Study- S.H., A.F.M.S.A., C.A., S.S.İ., A.Z.; Data Acquisition- S.H., A.F.M.S.A., C.A., S.S.İ., A.Z.; Data Analysis/Interpretation- S.H., A.F.M.S.A., C.A., S.S.İ., A.Z.; Drafting Manuscript- S.H., A.F.M.S.A., C.A., S.S.İ., A.Z.; Critical Revision of Manuscript- S.H., A.F.M.S.A., C.A., S.S.İ., A.Z.; Final Approval and Accountability- S.H., A.F.M.S.A., C.A., S.S.İ., A.Z.; Material and Technical Support- S.H., A.F.M.S.A., C.A., S.S.İ., A.Z.; Supervision- S.H., A.F.M.S.A., C.A., S.S.İ., A.Z.
Conflict of Interest: The authors have no conflict of interest to declare.
Grant Support: The authors declared that this study has received no financial support.

REFERENCES

Aamer N., Younus J., "A Performance Comparison of Data Encryption Algorithms", *First International Conference on IEEE Information and Communication Technologies (ICICT),* Vol 1, Issue 6, 27-28 Aug. 2005, pp 84-89.

Agrawal M., Mishra P., "A Comparative Survey on Symmetric Key Encryption Techniques" *International Journal on Computer Science and Engineering (IJCSE)*, Vol. 4 No. 05 May 2012, ISSN: 0975-3397.

Arya, E. S., & Kumar, A. (2017). Ascıı Based Encryptıon Decryptıon Technıque For Informatıon Securıty And Communıcatıon. In 3rd international conference on innovative trends in science, *Engineering and Management*, YMCA connaught place, New Delhi, 7th January.

Basharat, I., Azam, F. and Muzaffar, A., Database Security and Encryption: A Survey Study, *International Journal of Computer Applications*, 47(12): 28-34 (2012).

Bertino, E., & Sandhu, R. (2005). Database security-concepts, approaches, and challenges. *IEEE Transactions on Dependable and secure computing*, 2(1), 2-19.

Diaa E., Hatem K. and Mohie H., "Performance Evaluation of Symmetric Encryption Algorithms", *IJCSNS International Journal of Computer Science and Network Security,* VOL.8 No.12, pp. 280-286, December 2008.

Elminaam, D. S. A., Kader, H. M. A., & Hadhoud, M. M. (2009). Energy efficiency of encryption schemes for wireless devices. *International Journal of Computer Theory and Engineering,* 1(3), 302.

Emad F. Khalaf and Mustafa M. Kadi., "A Survey of Access Control and Data Encryption for Database Security", *JKAU: Eng. Sci*., Vol. 28 No. 1, pp: 19 - 30 (2017) Doi: 10.4197/Eng. 28-1.2.

Eskicioğlu, Ö. C. & Isık, A. H. (2022). Mobil Uyumlu Çoklu Dil Destekli Hibrit Şifreleme Algoritması . Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi , 24 (72) , 1007-1019 . DOI: 10.21205/deufmd.2022247228

Hamdan.O. Alanazi, B.B. Zaidan, A.A. Zaidan, Hamid A. Jalab, M. Shabbir and Y. Al-Nabhani, "New Comparative Study Between DES, 3DES and AES within Nine Factors" *Journal of Computing*, Volume 2, ISSUE 3, March 2010, ISSN 2151-9617.

Kakkar A. and Bansal P. K., "Reliable Encryption Algorithm used for Communication", *M. E. Thesis*, Thapar University, 2004.

Kaur, R., Kiranpreet and Verma, P., Survey on Database Security, *International Journal of Computer Applications*, 105(10): 27–31 (2014).

Kulkarni, S. and Urolagin, S., Review of Attacks on Databases and Database Security Techniques, *International Journal of Emerging Technology and Advanced Engineering*, 2(11): 253-263 (2012).

Luc B., Yanli GUO, "Database Encryption", *Article* · January 2010 DOI: 10.1007/978-1-4419-5906-5_677.

Mathur A., "A Research paper: An ASCII value based data encryption algorithm and its comparison with other symmetric data encryption algorithms", *International Journal on Computer Science and Engineering (IJCSE),* Vol. 4, pp. 1650-1657, Sep 2012 ISSN: 0975- 3397

Mittal H., "Optimized Encryption Algorithm using Dynamic Keys", *Electronics and Communication Engineering Department*, THAPAR UNIVERSITY PATIALA-147004, 2012

Nadeem A., "A performance comparison of data encryption algorithms," *IEEE Information and Communication Technologies*, pp. 84-89, 2006.

Patil, P., Narayankar, P., Narayan, D. G., & Meena, S. M. (2016). A comprehensive evaluation of cryptographic algorithms: DES, 3DES, AES, RSA and Blowfish. *Procedia Computer Science*, 78, 617-624.

Patil, A. and Meshram, B. B., Database Access Control Policies, *International Journal of Engineering Research and Applications,* 2(3): 3150–3154 (2012).

Rohilla, S. and Mittal, P. K., Database Security: Threats and Challenges, *International Journal of Advanced Research in Computer Science and Software Engineering,* 3(5): 810–813 (2013).

Sesay, S., Yang, Z., Chen, J., & Xu, D. (2005, January). A secure database encryption scheme. In Second *IEEE Consumer Communications and Networking Conference*, 2005. CCNC. 2005 (pp. 49-53). IEEE.

Seth, S. M., & Mishra, R. (2011). *Comparative analysis of encryption algorithms for data communication. IJCST Vol. 2, Issue 2, June 2, 2011.*

Shelly R., Pradeep Kumar M., Database Security: Threats and Challenges, International Journal of *Advanced Research in Computer Science and Software Engineering*, Volume 3, Issue 5, May 2013.

Shinge, S. R., & Patil, R. (2014). An encryption algorithm based on ASCII value of data. *International Journal of Computer Science and Information Technologies*, 5(6), 7232-7234.

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2002). *Database system concepts (Vol. 5).* New York: McGraw-Hill.

Singh, S. and Rai, R. K., A Review Report on Security Threats on Database*, International Journal of Computer Science and Information Technologies*, 5(3): 3215–3219 (2014).

Singh G., Supriya, "A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security" *International Journal of Computer Applications* (0975 – 8887) Volume 67– No.19, April 2013.

Thuraisingham, B., Database Security: Past, Present, and Future, *IEEE International Congress on Big Data*, 772–774 (2015).