Original Article

A COMPARATIVE SIMULATION OF KEY SIZE SECURITY IN RIJNDAEL, FPA, AND RC4 ENCRYPTION ALGORITHMS

[1]Hassan Abdullahi Mohammed, [2]Ibrahimah Khadijat Bello and [3]Sulaiman Ibrahim Kachalla

[1,2]Department of Computer Science,
Jigawa State Polytechnic, Dutse, Nigeria.
[3]Department of Computer Science, Umaru Ali Shinkafi Polytechnic, Sokoto, Nigeria.

DOI: https://doi.org/10.5281/zenodo.13902952

A breach of the unprotected electronic data might result in sensitive or confidential information being taken, changed, copied, sent, viewed, or used without authority. This study focuses on having security testing of Rijindael, Rivest Cipher4 and Format Preserving Algorithms in VB.Net Using simulation Comparison Techniques by comparing three (3) encryption algorithms by simulation method to know the most secured algorithm in VB.Net of different minimum and maximum key sizes to determine the security. The study also evaluates the three most common encryption algorithms like Rijandael, Rivest Cipher4 and Format Preserving Encryption Algorithm. A simulation comparison was conducted for those three (3) encryptions algorithms at different settings for each algorithm such as minimum and maximum key size change to know which among those algorithms performs cryptography in the most secured way. Results from simulation analysis showed that Rivest Cipher4 (RC4) is the best algorithm in terms of security among the other compared contenders using the higher key size and lower key size as it maintains an average time for each key size both (Min & Max) followed by Rijandael and format preserving algorithm. This study therefore recommends the adoption of Rivest Cipher4 (RC4) in VB.Net in terms of secured cryptography.

Keywords: Electronic Code Book, Format Preserving, Performance, Rijandael, Rivest Cipher, Simulation

Introduction

Syed el al. (2022) Data security is a method of guarding against unauthorized access to, corruption of, or theft of digital data at any stage of its existence. It is a concept that encompasses all elements of data security, including administrative and access controls, logical program security, and physical hardware and storage device security. Also included are organizational policies and practices (Zeebaree *et al*. 2019).

Hughes-Lartey *et al*. (2021) when properly implemented, comprehensive data security measures protect against insider threats and human error, which are still among the main reasons for data breaches in the modern day, as well as protecting an organization's information assets from cybercriminal activities. Giving the company better insight into the locations and uses of its critical data is necessary for the use of tools and technologies for data protection.

According to Michalas (2019), for each user (or group of users), file encryption with the user's key may be enabled, guaranteeing the confidentiality of personal documents even when the computer is shared because other users cannot decrypt and view text they lack permission to the keys. The backup is encrypted to produce ciphertext in order to further lessen the possibility of losing data caused by the theft or destruction of the backup medium. Although there are many secure file systems, Li *et al*. (2019) claim that the privacy of the entire system is compromised because key management security is either poorly understood or because key management security flaws exist.

Rijndael, also known as Advanced Encryption Standard (AES), is a symmetric key encryption algorithm known for its strong security and flexibility in key sizes (128, 192, and 256 bits). Research has shown that increasing the key size significantly enhances resistance to brute-force attacks, making Rijndael a preferred choice in many security applications (Rouhani & Deters (2019).

FPE algorithms allow data to be encrypted while maintaining its original format, which is critical for applications where the ciphertext must adhere to specific structural constraints. Studies indicate that the security of FPE can be influenced by key size, with larger keys generally providing better protection against unauthorized decryption attempts (Kaydos 2020).

Rivest Cipher 4 (RC4): Although once widely used for its speed and simplicity, RC4 has been scrutinized for vulnerabilities, particularly in key management and stream cipher attacks. Recent studies suggest that the effectiveness of RC4 can be compromised by small key sizes, highlighting the need for thorough security evaluations when implementing this algorithm Genc *et al*. (2020).

Given that most cryptographic research is conducted in C++ or Java, this research will provide new data on how these algorithms perform in the VB.NET environment. It will evaluate the three encryption algorithms Rijndael, Format Preserving Encryption, and RC4 under various settings, including changes in key sizes (both minimum and maximum), to determine which algorithm offers the highest level of security. This will address a clear gap for developers working in enterprise systems using the .NET framework.

Methodology

This study used simulation techniques in evaluating the security testing both (minimum and maximum) key sizes for Rijandael, Format Preserving Encryption and RC4 algorithms. The study also utilized VB.Net in making the simulation evaluation.

Analytical Instruments

The followings are the tools used for the algorithms analysis determined the most secured and fastest method.

Symmetric method: Any encryption technique that employs the same key for data encryption and decryption is referred to as symmetric, according to Alenezi *et al*. (2020). The Caesar Cipher is one of the easiest symmetric encryption techniques, yet it is also one of the easiest to crack.

Since then, several more symmetric encryption techniques have been created by cryptographers, including ones that are now used to protect information like keys.

VB.Net: Balaraju (2021) the system Security. The.NET Framework's cryptographic namespace offers a number of utilities to help with encryption and decrypting. One of the many classes offered is the CryptoStream class.

Original Article

Rijandael, Format Preserving and Rivest Cipher4 Security Testing of Key Sizes

Rijandael, Format Preserving, and RC4's security can be assessed in part by altering the key size and assessing how it affects the algorithm's performance. For instance, the three key sizes supported by the Rijandael standard are 128 bits, 192 bits, and 256 bits. Stronger security is often achieved by using larger keys, but this also necessitates more processing power and storage.

The following are the steps to assess the security of Rijandael, Format Preserving and RC4 using a key size change minimum/maximum using the developed VB.Net simulation:

- Selecting a test dataset: choose a dataset that reflects the kind of information you want to keep secure, such as audio, visual files or document.
- Use Rijandael, FPA or RC4 to encrypt the data with a predetermined key size: Depending on the degree of security you wish to accomplish, encrypt the dataset with AES using a key size of 128 bits, 192 bits, or 256 bits but the analysis program will take only minimum and maximum for the evaluation
- Calculate the encryption duration: Keep track of the amount of time it takes to encrypt the dataset with the chosen key size either minimum or maximum.
- Calculate the decryption time by decrypting the encrypted dataset with the same key size and noting how long it takes.
- Analyse the security parameter: To assess the algorithms security, compare the encryption and decryption timings for each key size.

The process has to be repeated to confirm the findings and make sure the key size chosen is adequate for particular use case. Repeat the process for various datasets and key sizes.

Mode of Executing Security Testing

The crucial component of encryption approaches that demonstrates how data is encrypted is key change management. This key length serves as the foundation for the encryption ratio. The variable key length used by the symmetric approach is longer. So, one of the three (3) algorithms that will be explored in the processing of encryption is key management. Each algorithm employs a particular amount of key lengths that are utilized as a process block cipher in all the parameters to determine the secured one.

The security testing was conducted using each algorithm (Rijandael. Format Preserving and Rivest Cipher) key size, the key size was used to determine the best secured algorithms.

The key change minimum and maximum size was calculated in guaranteeing the security analysis. The analysis consists of min/max key size, text, load and time which all these parameters are calculated by the key size to know the best secured algorithms. Each of the compared algorithms consists of more than three categories of key size where each key size was calculated and change according to time, load, and text cryptography, the algorithm that maintains maximum key size after a change in each activity is considered as the best.

Block Cipher Mode

By applying a cipher key to the whole block, the cipher block chaining (CBC) mode of a block cipher encrypts a set of bits as a single unit, or block. In cipher block chaining, an initialization vector (IV) of a particular length is employed. Using this conjunction with a single encryption key, as shown in figure 1 below, organizations and individuals may safely encrypt and decode enormous volumes of plaintext.

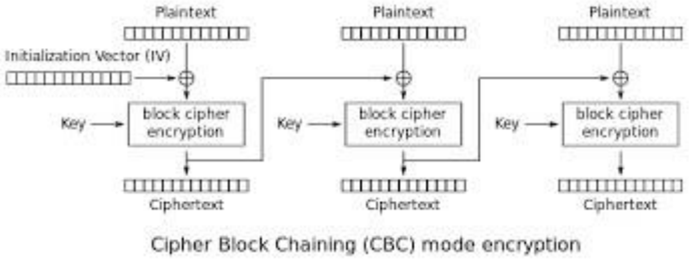


Figure 1: CBC Operation

Source: Mohammed *et al*. (2021)

Simulation Operations

The Simulation environment and the system components that were employed are covered in great depth in this section. Table 1 provides a comparison of the graphical user interface environment's performance and security for RC4, FPA, and Rijandael as well as the key size and block size for each technique. To give the algorithm with the greatest results, this implementation has undergone considerable testing and optimization.

The resolution improves the FPA, Rijandael, and RC4 managed wrappers for the System.

Table 1: Compared Decryption/Encryption Algorithm

S/No	Algorithms	Key Size in Bits	Block Size in Bits
1	RC4	64	128
2	FPA	192	64
3	Rijandael	128	192

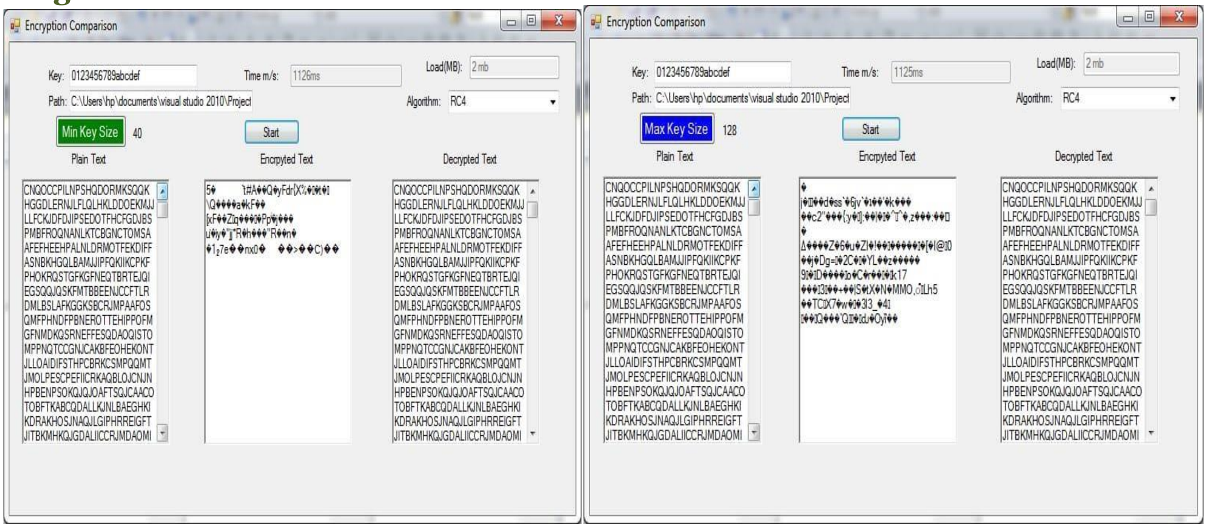
Source: Aleniri *et al*. (2021)

System Variables

The tests were conducted using an Intel Pentium Processor 3825U 64-bit processor and 4GB of RAM. The simulation software was developed utilizing VB.Net programming's default parameters. The experiments have to be repeated numerous times in order to guarantee that the findings are trustworthy and that they can be applied to assess other strategies.

Results

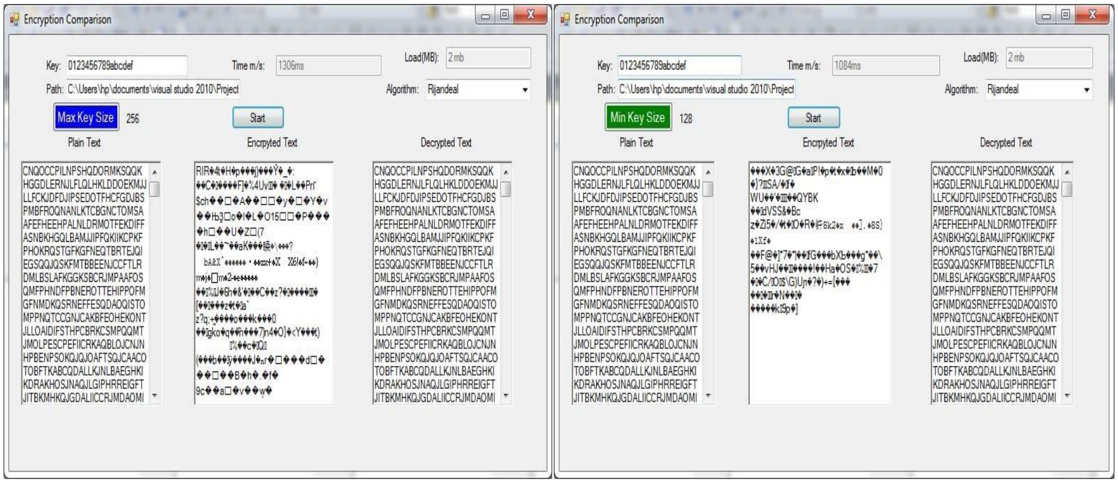
Below figures depicts the security key sizes evaluation for the Rijandael, Format Preserving and RC4 algorithms VB.Net simulation.



```
Key: 0123456789abcdef        Time m/s: 1126ms        Load(MB): 2 mb
Path: C:\Users\hp\documents\visual studio 2010\Project        Algorithm: RC4
Min Key Size 40

Plain Text                   Encrpyted Text                        Decrypted Text
CNQOCCPILNPSHQDORMKSQQK      5�      t#A��Q�yFdr{X%�t�t�          CNQOCCPILNPSHQDORMKSQQK
HGGDLERNJLFLQLHKLDDOEKMJJ    \Q����a�kF��                          HGGDLERNJLFLQLHKLDDOEKMJJ
LLFCKJDFDJIPSEDOTFHCFGDJBS   [xF��Zq���ɪ�Pp�j���                  LLFCKJDFDJIPSEDOTFHCFGDJBS
PMBFROQNANLKTCBGNCTOMSA      u�iy�"j"R�h���"R��n�                  PMBFROQNANLKTCBGNCTOMSA
AFEFHEEHPALNLDRMOTFEKDIFF    �1₂7e��nx0�    ��>��C)��              AFEFHEEHPALNLDRMOTFEKDIFF
ASNBKHGQLBAMJJIPFQKIIKCPKF                                          ASNBKHGQLBAMJJIPFQKIIKCPKF
PHOKRQSTGFKGFNEQTBRTEJQI                                            PHOKRQSTGFKGFNEQTBRTEJQI
EGSQQJQSKFMTBBEENJCCFTLR                                            EGSQQJQSKFMTBBEENJCCFTLR
DMLBSLAFKGGKSBCRJMPAAFOS                                            DMLBSLAFKGGKSBCRJMPAAFOS
QMFPHNDFPBNEROTTEHIPPOFM                                            QMFPHNDFPBNEROTTEHIPPOFM
GFNMDKQSRNEFFESQDAOQISTO                                            GFNMDKQSRNEFFESQDAOQISTO
MPPNQTCCGNJCAKBFEOHEKONT                                            MPPNQTCCGNJCAKBFEOHEKONT
JLLOAIDIFSTHPCBRKCSMPQQMT                                           JLLOAIDIFSTHPCBRKCSMPQQMT
JMOLPESCPEFIICRKAQBLOJCNJN                                          JMOLPESCPEFIICRKAQBLOJCNJN
HPBENPSOKQJQJOAFTSQJCAACO                                           HPBENPSOKQJQJOAFTSQJCAACO
TOBFTKABCQDALLKJNLBAEGHKI                                           TOBFTKABCQDALLKJNLBAEGHKI
KDRAKHOSJNAQJLGIPHRREIGFT                                           KDRAKHOSJNAQJLGIPHRREIGFT
JITBKMHKQJGDALIICCRJMDAOMI                                          JITBKMHKQJGDALIICCRJMDAOMI
```

```
Key: 0123456789abcdef        Time m/s: 1125ms        Load(MB): 2 mb
Path: C:\Users\hp\documents\visual studio 2010\Project        Algorithm: RC4
Max Key Size 128

Plain Text                   Encrpyted Text                              Decrypted Text
CNQOCCPILNPSHQDORMKSQQK      �                                           CNQOCCPILNPSHQDORMKSQQK
HGGDLERNJLFLQLHKLDDOEKMJJ    j�▯��d�ss`�6jv`�ı��'�k���                 HGGDLERNJLFLQLHKLDDOEKMJJ
LLFCKJDFDJIPSEDOTFHCFGDJBS   ��c2"���{.y�▯]:��|�▯�^▯^�,z���:��▯         LLFCKJDFDJIPSEDOTFHCFGDJBS
PMBFROQNANLKTCBGNCTOMSA      �                                           PMBFROQNANLKTCBGNCTOMSA
AFEFHEEHPALNLDRMOTFEKDIFF    Δ����Z�6�u�ZI�!��▯������▯�[�I@▯0           AFEFHEEHPALNLDRMOTFEKDIFF
ASNBKHGQLBAMJJIPFQKIIKCPKF   ��j�Dg=▯�2C�▯�YL��z�����                    ASNBKHGQLBAMJJIPFQKIIKCPKF
PHOKRQSTGFKGFNEQTBRTEJQI     9▯�▯D����▯b�C�r��▯�▯k17                    PHOKRQSTGFKGFNEQTBRTEJQI
EGSQQJQSKFMTBBEENJCCFTLR     ���▯3▯��+��|S�tX�N�MMO,ŏ▯Lh5               EGSQQJQSKFMTBBEENJCCFTLR
DMLBSLAFKGGKSBCRJMPAAFOS     ��TC▯X7�w�▯�3I3_�4▯                         DMLBSLAFKGGKSBCRJMPAAFOS
QMFPHNDFPBNEROTTEHIPPOFM     ▯��▯Q���`Q▯▯�▯dú�Oyĩ��                      QMFPHNDFPBNEROTTEHIPPOFM
GFNMDKQSRNEFFESQDAOQISTO                                                 GFNMDKQSRNEFFESQDAOQISTO
MPPNQTCCGNJCAKBFEOHEKONT                                                 MPPNQTCCGNJCAKBFEOHEKONT
JLLOAIDIFSTHPCBRKCSMPQQMT                                                JLLOAIDIFSTHPCBRKCSMPQQMT
JMOLPESCPEFIICRKAQBLOJCNJN                                               JMOLPESCPEFIICRKAQBLOJCNJN
HPBENPSOKQJQJOAFTSQJCAACO                                                HPBENPSOKQJQJOAFTSQJCAACO
TOBFTKABCQDALLKJNLBAEGHKI                                                TOBFTKABCQDALLKJNLBAEGHKI
KDRAKHOSJNAQJLGIPHRREIGFT                                                KDRAKHOSJNAQJLGIPHRREIGFT
JITBKMHKQJGDALIICCRJMDAOMI                                               JITBKMHKQJGDALIICCRJMDAOMI
```

Figure 2: GUI Simulation of RC4 algorithm minimum and maximum key size encrypt/decrypt testing

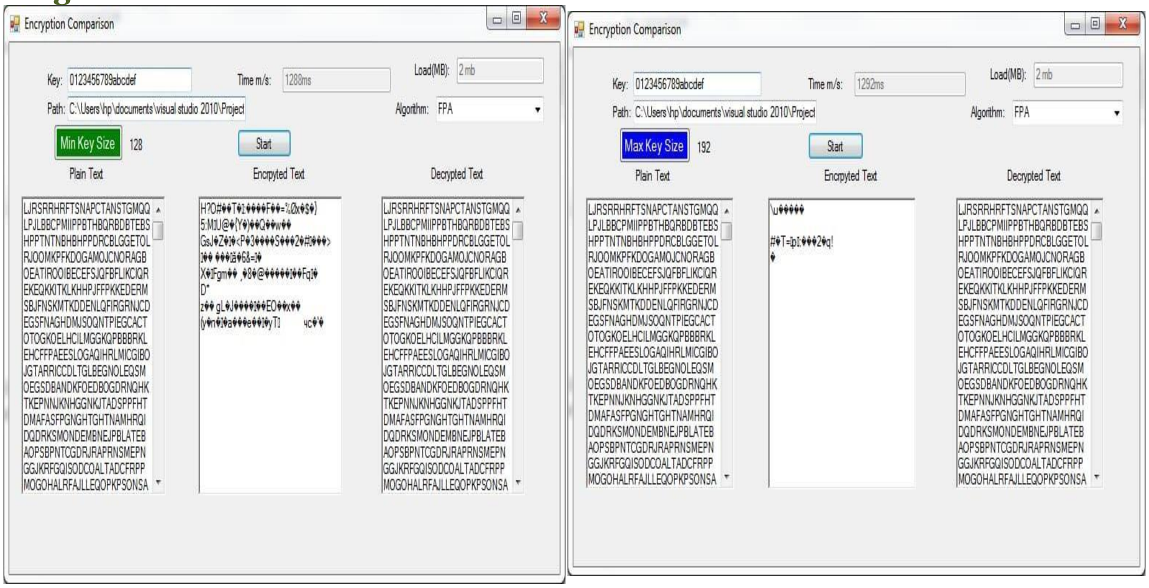


```
Key: 0123456789abcdef        Time m/s: 1306ms        Load(MB): 2 mb
Path: C:\Users\hp\documents\visual studio 2010\Project        Algorithm: Rijandeal
Max Key Size 256

Plain Text                   Encrpyted Text                              Decrypted Text
CNQOCCPILNPSHQDORMKSQQK      RIR�4t�H�p���j)���Ý�_�:                     CNQOCCPILNPSHQDORMKSQQK
HGGDLERNJLFLQLHKLDDOEKMJJ    ��C�▯����F]�%4Uv▯▯� �▯�L��Prr              HGGDLERNJLFLQLHKLDDOEKMJJ
LLFCKJDFDJIPSEDOTFHCFGDJBS   Sch��□�A��□□�y�□�Y�v                      LLFCKJDFDJIPSEDOTFHCFGDJBS
PMBFROQNANLKTCBGNCTOMSA      ��Hb3□o�I�L�O15□□�P���                     PMBFROQNANLKTCBGNCTOMSA
AFEFHEEHPALNLDRMOTFEKDIFF    �h□��U�Z□(7                                AFEFHEEHPALNLDRMOTFEKDIFF
ASNBKHGQLBAMJJIPFQKIIKCPKF   �▯�▯L��~��aK����\���?                      ASNBKHGQLBAMJJIPFQKIIKCPKF
PHOKRQSTGFKGFNEQTBRTEJQI       bA&X^������·��zzc+�X   X6!�f~��}          PHOKRQSTGFKGFNEQTBRTEJQI
EGSQQJQSKFMTBBEENJCCFTLR     m�j�□m�2-�e�����                           EGSQQJQSKFMTBBEENJCCFTLR
DMLBSLAFKGGKSBCRJMPAAFOS     ��▯%▯J�6h�&'�▯��C��z?�▯����▯▯�            DMLBSLAFKGGKSBCRJMPAAFOS
QMFPHNDFPBNEROTTEHIPPOFM     [��▯���z�t�▯a`                            QMFPHNDFPBNEROTTEHIPPOFM
GFNMDKQSRNEFFESQDAOQISTO     z?q;+����o���k���▯                         GFNMDKQSRNEFFESQDAOQISTO
MPPNQTCCGNJCAKBFEOHEKONT     ��▯gko�q��ñ���7)n4�O}�<Y���t)             MPPNQTCCGNJCAKBFEOHEKONT
JLLOAIDIFSTHPCBRKCSMPQQMT             ▯%��c�▯Q▯                         JLLOAIDIFSTHPCBRKCSMPQQMT
JMOLPESCPEFIICRKAQBLOJCNJN   (���b��▯y����J�nr�□����d□�                JMOLPESCPEFIICRKAQBLOJCNJN
HPBENPSOKQJQJOAFTSQJCAACO    ��□��B�h�.�f�                              HPBENPSOKQJQJOAFTSQJCAACO
TOBFTKABCQDALLKJNLBAEGHKI    9c��a□�v��w�                               TOBFTKABCQDALLKJNLBAEGHKI
KDRAKHOSJNAQJLGIPHRREIGFT                                                KDRAKHOSJNAQJLGIPHRREIGFT
JITBKMHKQJGDALIICCRJMDAOMI                                               JITBKMHKQJGDALIICCRJMDAOMI
```

```
Key: 0123456789abcdef        Time m/s: 1084ms        Load(MB): 2 mb
Path: C:\Users\hp\documents\visual studio 2010\Project        Algorithm: Rijandeal
Min Key Size 128

Plain Text                   Encrpyted Text                              Decrypted Text
CNQOCCPILNPSHQDORMKSQQK      ���X�3G@▯G�a▯P!�p�t�x�▯b��M�0               CNQOCCPILNPSHQDORMKSQQK
HGGDLERNJLFLQLHKLDDOEKMJJ    �}?▯▯SA/�▯f�                               HGGDLERNJLFLQLHKLDDOEKMJJ
LLFCKJDFDJIPSEDOTFHCFGDJBS   WU��'�▯▯��QYBK                             LLFCKJDFDJIPSEDOTFHCFGDJBS
PMBFROQNANLKTCBGNCTOMSA      ��▯dVSS&�Bc                                PMBFROQNANLKTCBGNCTOMSA
AFEFHEEHPALNLDRMOTFEKDIFF    z�Zi5�/�t�▯O�R�▯8k2�x  ��].�8S)            AFEFHEEHPALNLDRMOTFEKDIFF
ASNBKHGQLBAMJJIPFQKIIKCPKF   �1Xf�                                      ASNBKHGQLBAMJJIPFQKIIKCPKF
PHOKRQSTGFKGFNEQTBRTEJQI     ��F@�]"7�"|��▯fG���bXb���g"��\            PHOKRQSTGFKGFNEQTBRTEJQI
EGSQQJQSKFMTBBEENJCCFTLR     5��vHJ��▯▯����!��Ha�OS�▯%▯▯�7             EGSQQJQSKFMTBBEENJCCFTLR
DMLBSLAFKGGKSBCRJMPAAFOS     �▯�C/▯O▯$\G)Uл�?�)+={���                   DMLBSLAFKGGKSBCRJMPAAFOS
QMFPHNDFPBNEROTTEHIPPOFM     ��▯�▯▯r�N��▯�                              QMFPHNDFPBNEROTTEHIPPOFM
GFNMDKQSRNEFFESQDAOQISTO     �����k▯9p�]                                GFNMDKQSRNEFFESQDAOQISTO
MPPNQTCCGNJCAKBFEOHEKONT                                                 MPPNQTCCGNJCAKBFEOHEKONT
JLLOAIDIFSTHPCBRKCSMPQQMT                                                JLLOAIDIFSTHPCBRKCSMPQQMT
JMOLPESCPEFIICRKAQBLOJCNJN                                               JMOLPESCPEFIICRKAQBLOJCNJN
HPBENPSOKQJQJOAFTSQJCAACO                                                HPBENPSOKQJQJOAFTSQJCAACO
TOBFTKABCQDALLKJNLBAEGHKI                                                TOBFTKABCQDALLKJNLBAEGHKI
KDRAKHOSJNAQJLGIPHRREIGFT                                                KDRAKHOSJNAQJLGIPHRREIGFT
JITBKMHKQJGDALIICCRJMDAOMI                                               JITBKMHKQJGDALIICCRJMDAOMI
```

Figure 3: GUI VB.Net Simulation of Rijandael algorithm maximum and minimum key size encrypt/decrypt testing

Original Article



Encryption Comparison

Key: 0123456789abcdef
Time m/s: 1288ms
Load(MB): 2 mb
Path: C:\Users\hp\documents\visual studio 2010\Project
Algorithm: FPA
Min Key Size 128

```
Plain Text                    Encrpyted Text                   Decrypted Text
LJRSRRHRFTSNAPCTANSTGMQQ      H?O#◆◆T◆◆◆◆◆F◆◆=%Øx◆$◆}          LJRSRRHRFTSNAPCTANSTGMQQ
LPJLBBCPMIIPPBTHBQRBDBTEBS    5:M◆U@◆{Y◆}◆◆Q◆◆w◆◆              LPJLBBCPMIIPPBTHBQRBDBTEBS
HPPTNTNBHBHPPDRCBLGGETOL      GsJ◆Z◆◆<P◆3◆◆◆◆S◆◆◆2◆#◆◆◆>       HPPTNTNBHBHPPDRCBLGGETOL
RJOOMKPFKDOGAMOJCNORAGB       ◆◆ ◆◆◆◆◆6&=◆◆                    RJOOMKPFKDOGAMOJCNORAGB
OEATIROOIBECEFSJQFBFLIKCIQR   X◆◆Fgm◆◆ ,◆8◆@◆◆◆◆◆◆◆Fq◆◆         OEATIROOIBECEFSJQFBFLIKCIQR
EKEQKKITKLKHHPJFFPKKEDERM     D*                               EKEQKKITKLKHHPJFFPKKEDERM
SBJFNSKMTKDDENLQFIRGRNJCD     z◆◆ gL◆J◆◆◆◆◆◆◆EO◆◆x◆◆           SBJFNSKMTKDDENLQFIRGRNJCD
EGSFNAGHDMJSOQNTPIEGCACT      {y◆n◆◆◆a◆◆◆e◆◆◆yT◆     чc◆'◆     EGSFNAGHDMJSOQNTPIEGCACT
OTOGKOELHCILMGGKQPBBBRKL                                       OTOGKOELHCILMGGKQPBBBRKL
EHCFFPAEESLOGAQIHRLMICGIBO                                     EHCFFPAEESLOGAQIHRLMICGIBO
JGTARRICCDLTGLBEGNOLEQSM                                       JGTARRICCDLTGLBEGNOLEQSM
OEGSDBANDKFOEDBOGDRNQHK                                        OEGSDBANDKFOEDBOGDRNQHK
TKEPNNJKNHGGNKJTADSPPFHT                                       TKEPNNJKNHGGNKJTADSPPFHT
DMAFASFPGNGHTGHTNAMHRQI                                        DMAFASFPGNGHTGHTNAMHRQI
DQDRKSMONDEMBNEJPBLATEB                                        DQDRKSMONDEMBNEJPBLATEB
AOPSBPNTCGDRJRAPRNSMEPN                                        AOPSBPNTCGDRJRAPRNSMEPN
GGJKRFGQISODCOALTADCFRPP                                       GGJKRFGQISODCOALTADCFRPP
MOGOHALRFAJLLEQOPKPSONSA                                       MOGOHALRFAJLLEQOPKPSONSA
```

Encryption Comparison

Key: 0123456789abcdef
Time m/s: 1292ms
Load(MB): 2 mb
Path: C:\Users\hp\documents\visual studio 2010\Project
Algorithm: FPA
Max Key Size 192

```
Plain Text                    Encrpyted Text      Decrypted Text
LJRSRRHRFTSNAPCTANSTGMQQ      \u◆◆◆◆◆             LJRSRRHRFTSNAPCTANSTGMQQ
LPJLBBCPMIIPPBTHBQRBDBTEBS                        LPJLBBCPMIIPPBTHBQRBDBTEBS
HPPTNTNBHBHPPDRCBLGGETOL      #◆T=p◆◆◆◆2◆q!       HPPTNTNBHBHPPDRCBLGGETOL
RJOOMKPFKDOGAMOJCNORAGB       ◆                   RJOOMKPFKDOGAMOJCNORAGB
OEATIROOIBECEFSJQFBFLIKCIQR                       OEATIROOIBECEFSJQFBFLIKCIQR
EKEQKKITKLKHHPJFFPKKEDERM                         EKEQKKITKLKHHPJFFPKKEDERM
SBJFNSKMTKDDENLQFIRGRNJCD                         SBJFNSKMTKDDENLQFIRGRNJCD
EGSFNAGHDMJSOQNTPIEGCACT                          EGSFNAGHDMJSOQNTPIEGCACT
OTOGKOELHCILMGGKQPBBBRKL                          OTOGKOELHCILMGGKQPBBBRKL
EHCFFPAEESLOGAQIHRLMICGIBO                        EHCFFPAEESLOGAQIHRLMICGIBO
JGTARRICCDLTGLBEGNOLEQSM                          JGTARRICCDLTGLBEGNOLEQSM
OEGSDBANDKFOEDBOGDRNQHK                           OEGSDBANDKFOEDBOGDRNQHK
TKEPNNJKNHGGNKJTADSPPFHT                          TKEPNNJKNHGGNKJTADSPPFHT
DMAFASFPGNGHTGHTNAMHRQI                           DMAFASFPGNGHTGHTNAMHRQI
DQDRKSMONDEMBNEJPBLATEB                           DQDRKSMONDEMBNEJPBLATEB
AOPSBPNTCGDRJRAPRNSMEPN                           AOPSBPNTCGDRJRAPRNSMEPN
GGJKRFGQISODCOALTADCFRPP                          GGJKRFGQISODCOALTADCFRPP
MOGOHALRFAJLLEQOPKPSONSA                          MOGOHALRFAJLLEQOPKPSONSA
```

Figure 4: GUI VB.Net Simulation of format preserving algorithm minimum key size decrypt/encrypt testing

Table 2: Results of the VB.Net Rijandael, Format Preserving and Rivest Cipher 4 Algorithms

S/No.	Algorithm/Min and Max Key Size	Encrypt/Decrypt Time Taken Using Minimum Key Size	Encrypt/Decrypt Time Taken Using Maximum Key Size
1.	Rijandael (Max 256 and Min 128)	1306MS in 2MB Data Load	1084MS in 2MB Data Load
2.	Rivest Cipher 4 (Max 128 and Min 40)	1126MS in 2MB Data Load	1125MS in 2MB Data Load
3.	Format Preserving Algorithm (Min 128 and Max 192)	1288MS in 2MB Data Load	1292MS in 2MB Data Load

Table 2 above shows that RC4 performs very well using the maximum key size and minimum key size as it maintains an average time for each key size both (Min & Max) followed by Rijandael and format preserving algorithm.

The simulation result as depicted in figure 5 below shows that Rivest Cipher4 (RC4) is the best algorithm among the other two contenders using higher key size and lower key size as it maintain an average time for each key size (Minimum and Maximum) followed by Rijandael second to RC4, the minimum key size used in Rijandael Algorithm is second to known and by far would have been the best algorithm but for it maximum key size is worse as it takes time to finished but FPA is not far worse algorithm but it takes time to finished than other algorithms with average worst case scenario.

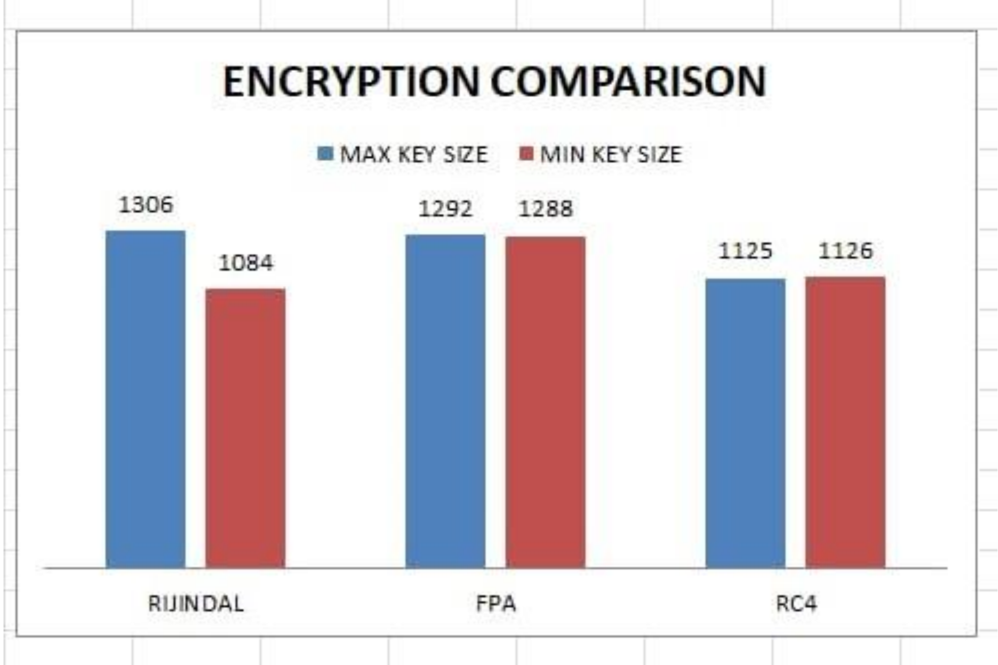


Figure 5: Result of the VB.Net simulation of Rijandael, FPA and RC4

Discussion

The Rijandael Algorithm, Format Preserving Algorithm, and Rivest Cipher4 have been examined in this study to see which one is more secure for preventing unauthorized access to data using Minimum and Maximum Algorithms key sizes.

The methods' security in the cryptography time was assessed using minimum and maximum key size, allowing the most secure one in VB.Net to be identified. The strength of a cryptographic key is determined by the key size. The key size, which is expressed in bits, is represented using the binary number system. The key size may differ depending on the applications and cryptographic algorithm being used; for instance, it may be 40 bits, 56 bits, 128 bits, and so on. To protect the cipher text from the brute-force attack, the key size should be sufficiently enough that the attacker cannot decrypt it in a predefined amount of time. For such comparisons, higher key size and lower key size is utilized.

Rivest Cipher4 (RC4), which maintains an average time for each key size (Minimum and Maximum), is the best algorithm among the two competitors using higher key sizes and lower key sizes, as shown by the comparison results by simulation technique shown in figure 5. Rijandael, which came in second place, uses a minimum key size that is second to known and by far would have been the best algorithm, but its maximum key size is worse.

Evaluation Security Performance by Alenezi *et al*. (2020) argued that Rivest Cipher4 and Blowfish algorithm appears to be the best in terms of security and time fastness as the algorithms don't take much time while performing cryptography operations. They added that the two algorithms are better because of the accuracy, as the load increases the time decreases this amount to time maintaining.

A study by Taopana *et al*. (2019) supports the use of the encryption algorithm known as format preserving, which keeps the information's format intact while it is being encrypted. Although FPE is less effective than the industry standard Advanced Encryption Standard (AES), it can nevertheless maintain the format and length of data.

Following the simulation techniques result as depicted above in figure 5, Rivest Cipher4 is recommended for adoption as the most secured one in VB.Net cryptography as it maintain average time of cryptography in VB.Net programming Language.

References

Alenezi, M. N., Alabdulrazzaq, H., & Mohammad, N. Q. (2020). Symmetric encryption algorithms: Review and evaluation study. *International Journal of Communication Networks and Information Security*, *12*(2), 256-272.

Balaraju, J. (2021). Design and analysis of secure hadoop clusters using DNA cryptography.

Genc, T. S., & Raju, Y. D. S. (2020). Implementation of Data Security with Wallace Tree Approach Using Elliptical Curve Cryptography on FPGA. *Turkish Journal of Computer and Mathematics Education* (TURCOMAT), 12(6), 1546-1553.

Hughes-Lartey, K., Li, M., Botchey, F. E., & Qin, Z. (2021). Human factor, a critical weak point in the information security of an organization's Internet of things. *Heliyon*, *7*(3), e06522.

Kaydos, W. (2020). *Operational performance measurement: increasing total productivity*. CRC press.

Krombholz, K. (2021). On the Usability of Authenticity Checks for Hardware Security Tokens. *In 30th {USENIX} Security Symposium* ({USENIX} Security 21).

Li, D., Cai, Z., Deng, L., Yao, X., & Wang, H. H. (2019). The Information Security Model of Blockchain Is Based on Intrusion Sensing In The IoT Environment. *Cluster Computing*, 22(1), 451-468.

Li, D., Deng, L., Lee, M., & Wang, H. (2019). IoT Data Feature Extraction And Intrusion Detection System For Smart Cities Based on Deep Migration Learning. *International Journal of Information Management*, 49, 533-545.

Li, J., Greenwood, D., & Kassem, M. (2019). Blockchain in the Built Environment and Construction Industry: A Systematic Review, Conceptual Models and Practical Use Cases. *Automation in Construction*, *102*, 288-307.

Liang, Y., He, F., & Li, H. (2019). An Asymmetric and Optimized Encryption Method to Protect the Confidentiality of the 3D Mesh Model. *Advanced Engineering Informatics*, 42, 100963.

Liestyowati, D. (2020, March). Public key cryptography. In *Journal of Physics: Conference Series* (Vol. 1477, No. 5, p. 052062). IOP Publishing.

Logunleko, K. B., Adeniji, O. D., & Logunleko, A. M. (2020). A comparative study of symmetric cryptography mechanism on DES AES and EB64 for information security. *Int. J. Sci. Res. in Computer Science and Engineering*, *8*(1).

Makeri, Y. A. (2020). Integrated Cryptographical Access Control Over Network Project. *Acta Informatica Malaysia (AIM)*, *4*(1), 19-21.

Michalas, A. (2019, April). The Lord of the Shares: Combining Attribute-Based Encryption and Searchable Encryption for Flexible Data Sharing. *In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing* (pp. 146-155).

Mohammed, S. J., & Taha, D. B. (2022, March). Performance Evaluation of RSA, ElGamal, and Paillier Partial Homomorphic Encryption Algorithms. In *2022 International Conference on Computer Science and Software Engineering (CSASE)* (pp. 89-94). IEEE.

Rouhani, S., & Deters, R. (2019). Security, performance, and applications of smart contracts: A systematic survey. *IEEE Access*, *7*, 50759-50779.

Syed, N. F., Shah, S. W., Trujillo-Rasua, R., & Doss, R. (2022). Traceability in supply chains: A Cyber security analysis. *Computers & Security*, *112*, 102536.

Zeebaree, S. R., Zebari, R. R., Jacksi, K., & Hasan, D. A. (2019). Security Approaches For Integrated Enterprise Systems Performance: A Review. *Int. J. Sci. Technol. Res*, *8*(12).