Performance Analysis of Lightweight and Twofish Encryption Algorithm

Dr. Swapna Borde[1], Shashank Joshi[2], Saili Manchkar[3]
Department of Computer Engineering
[1, 2, 3] Vidyavardhini's College of Engineering and Technology, Vasai, Palghar, India

Abstract-*In today's scenario, Information Security is the most challenging aspects in the web and network application. Data security includes protecting the data, from pernicious forces and from the unwanted actions of unauthorized users. Thus, modern data communication uses cryptography which is an effective, powerful, and crucial component for safe or secure transmission of information by implementing security framework counting confidentiality, authentication, accountability, and accuracy in order to handle the security hazards. Data encryption has always been the last priority in mobile devices mainly due to the shallow knowledge of security by the users and limited resources of the smartphones. This security mechanism includes some algorithms to transform the data into indecipherable text called cipher-text. The legitimate users who possess the associated keys can decrypt it. A hardware implementation is restricted because of embedding the encryption algorithms in other applications. Thus a major cause of overall degraded performance of the system. So selecting right encryption algorithm in your application is of great concern. This paper implements two secret key encryption algorithms i.e. Light weight and Twofish and performs comparative analysis considering certain parameters.*

Keywords-Encryption, Lightweight, Twofish, Performance analysis.

I. INTRODUCTION

Mobile devices such as smart phones and tablets have caused an exemplary shift into nearly every field of the computing industry over a few decades era. It has been observed that the android smartphones has included the ability for whole disk encryption [2], users have been exposed to many existing solutions to encrypt the data and individual file locally, or to encrypt data that is to be stored or transmitted securely. However, these contribute to essential encryption methods primarily designed for systems where resources such as battery life and memory footprint are less constrained than other devices. Thus taking into account the resource constrictions of battery powered mobile devices, it is obvious that a reliable encryption scheme is required to protect and secure the information and also to alleviate the threat of data theft in future.

Cryptography is the practice and study of techniques which incorporates the ideas and methods of transforming a comprehensible message into one that is ambiguous, and then retransforming that message back into its original form. It has existed for thousands of years and it assures security and authentication of data [fig.1]. Cryptography can be furthercategorized as symmetric key cryptography and asymmetric key cryptography [3].

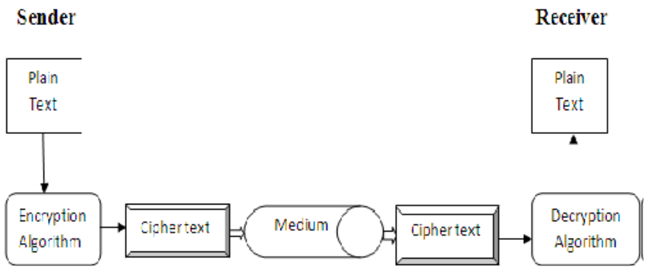


Figure 1. Cryptographic Process

Further, to evaluate the proposed method, it is implemented on Java platform with the Android SDK to develop an android application. We have used Lava icon with Android 4.4.2, 1GB RAM, Non-removable Li-Ion 4325 mAh battery (16 Wh). Trepn Profiler is a diagnostic tool to profile the performance and power consumption of android applications running on mobile devices. Application used in this project is Qualcomm Trepn Profiler1. Objective of this paper is to perform comparative analysis using various parameters and to select the best encryption scheme in terms of power and space used while maintaining the proper functionality of the system.

II. PROPOSED ALGORITHM

A. Twofish algorithm

Twofish was designed by Bruce Schneier, John Kelsey, Doug Withing, David Wagner, Chris Hall, and Niels Ferguson. It is asymmetric key block cipher with keys of variable length. Twofish is considered to be one of the traditional block cipher and is the successor of Blowfish. It contains 2 main diffusion elements, they are MDS matrix and PHT [4]. Our proposed system uses 128 bit key size to encrypt and decrypt a text of 128 bit block size. Twofish contains a 16-round Feistel structure like DES.

Twofish originated from an attempt to take the original Blowfish design and modify it for a 128-bit block. It is not designed close to the edge. In other words, leave a margin for error and provide more security than is provably required. Also, try to design against attacks that are not yet known. A guiding design principle behind Twofish is that the round function should be simple enough for us to keep in our heads.

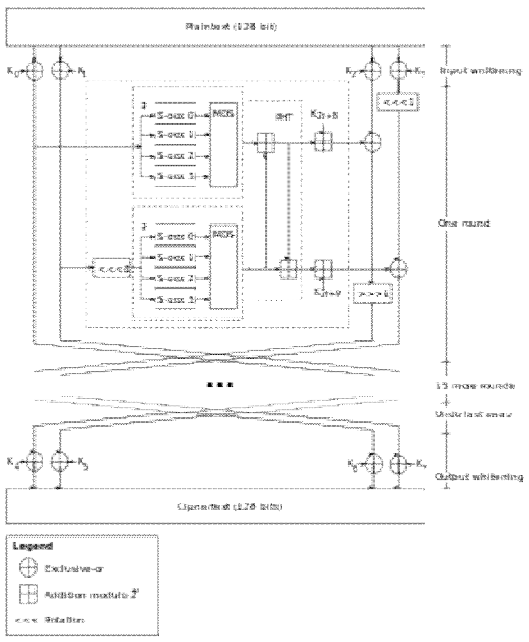


Figure 2: Twofish Encryption Algorithm

1. F - Function

The Feistel function F is a key-dependent permutation on 64 bit values. It takes three arguments, two input words P0 and P1, and the round number r is used to select the appropriate sub keys. The inputs are then passed to G-function and PHT.

i. G – Function

The function g forms the heart of Twofish. It consists of S-box and MDS block .An S-box is a table-driven non-linear operation built by using two fixed 8-by-8-bit permutations and key material. A maxi2mum distance separable (MDS) code is a linear mapping from a field elements to b field elements, producing a composite vector.

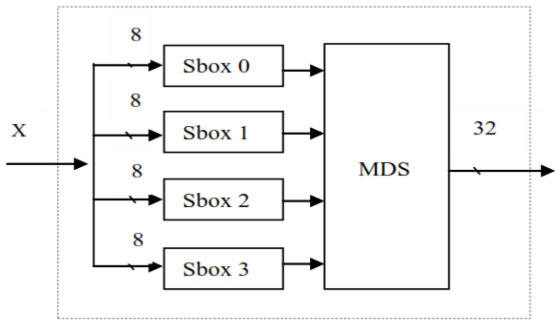


Figure 3: G-Function

Working of G-Function is as follows:

1. The input word X is split into four bytes.
2. Each byte is run through its own key-dependent S-box. Each S box takes 8 bits of input, and produces 8 bits of output.
3. The result obtained from S-box are then interpreted as a vector of length 4, and multiplied by the 4×4 MDS matrix.
4. The resulting vector is interpreted as a 32-bit word which is the result of g.

$$\begin{pmatrix} z0 \\ z1 \\ z2 \\ z3 \end{pmatrix} = \begin{pmatrix} 01 & EF & 5B & 5B \\ 5B & EF & EF & 01 \\ EF & 5B & 01 & EF \\ EF & 01 & EF & 5B \end{pmatrix} X \begin{pmatrix} y0 \\ y1 \\ y2 \\ y3 \end{pmatrix}$$

Result of g function MDS matrix Result of s boxes

Figure 4: Working of G-function

ii. PHT

Pseudo-Hadamard transform (PHT) is a reversible transformation of a bit string that provides cryptographic diffusion. In this, addition operation is performed. Twofish uses a 32-bit PHT to mix the outputs from its two parallel 32-bit g functions.

For the inputs a and b, the 32-bit PHT performs following operation:

a' = a + b mod 2^{32}

b'= a + 2b mod 2^{32}

2. Whitening

Whitening was used by Merkle in Khufu/Khafre, and independently invented by Rivest for DES-X [8]. Whitening is the technique of XORing key material before the first round and after the last round. Whitening increases the diffculty of key search attacks [8].

III. ALGORITHM

For Encryption

1. 128 bit input plain-text is divided into four parts of 32 bits each and is XORed with four 32 bits sub-keys.
2. The result obtained is then passed into F-Function for further computation.
3. The whole procedure is repeated 16 times.
4. Outputs obtained are further XORed with 4 more sub-keys.
5. The resultant output obtained after output whitening is 128-bit cipher-text.

The Decryption procedure is done in the same way by reversing the order of sub-keys.

B. Lightweight Algorithm

Lightweight encryption scheme consists of substitution, permutation and rotation. It is explicitly implemented for the Android mobile platform. Its design is best suited to provide encryption to the mobile based platform and reduces the overhead caused by several traditional encryption schemes. This algorithm is supposed to be a step to bridge the increasing personal and business demands for mobile security without having a pernicious impact to resources and performance of mobile systems. It enables encryption of any size of file since it operates at byte level of data.

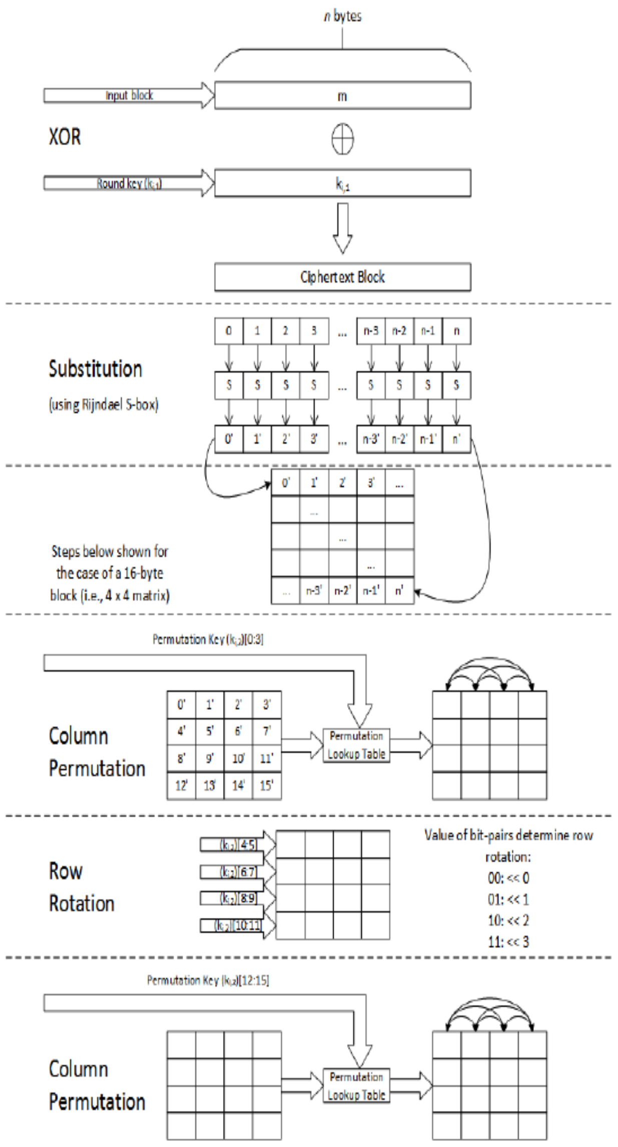


Figure 5. Lightweight Encryption algorithm

1. Proposed key generation steps:

This paper specifies the technique to derive a key using Password based key derivation function (PBKDF) that only the authorized user knows and something that only he has. The key generation is based on 2 factors: user accreditation and the unique IMEI number of the device [5].The resultant 1008-bit key is broken into 144-bit keys for seven rounds each, which is further categorized into 2 parts:

i. 128-bit XOR key
ii. 16-bit permutation key [2].

The main idea of a PBKDF is to slow dictionary or brute force attacks on the passwords by increasing the time needed to test each password.

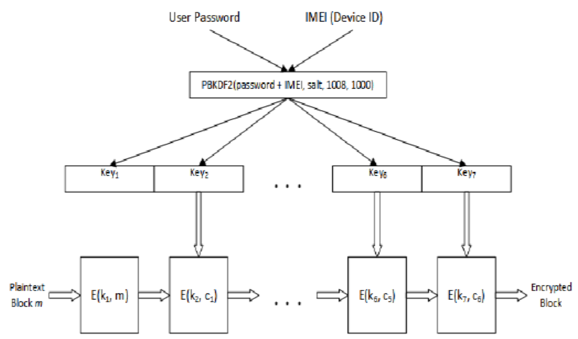


Figure 6 .Architecture of the round-based encryption scheme.

2. Steps of proposed algorithm

i. Cipher Block Chaining (CBC)

In this operation mode, each of the plaintext block is mandatorily XORed with that of the previous cipher-text block and then the result is encrypted with the key. It is beneficial for to the applications which require both symmetric encryption and data origin authentication.

ii. Substitution

Each byte from the XORed block is replaced using Basic lookup for a new byte. The resultant cipher text containing the substituted or replaced bytes is treated as a 4 x 4 matrix [6].

iii. Column permutation

In the first column permutation, first four bits of 16-bit permutation key and the lookup table demonstrated in fig.4 are used together to permute the columns of 4 x 4 matrix. Each column of the resultant matrix is permuted again using last four bits of permutation key in the second column permutation.

	00	01	10	11
00xx	1243	2341	3412	1342
01xx	4231	2314	4312	3421
10xx	4132	4123	1234	2413
11xx	3124	1423	3241	2134

Table 1. Basic Lookup table

iv. Row rotation

This is a fundamental concept that enables to determine the amount by which the row gets rotated to the left. In this, the next 8-bit are split among the 4 rows having 2 per row.

v. Padding

In a current block, if the number of bytes read is less than sixteen blocks then the method pads the block with desired number of bytes and continues with the encryption process. During decryption, the last bytes of the last blocks are examined and appropriate bytes are removed.

IV. ALGORITHM

For encryption:

1.128-bit text is XORed with 128-bit XOR key to produce 128-bit resultant cipher-text.

2. Substitution is performed on the cipher-text which results in 4 x 4 matrix.

3. Permute the columns using first 4 bits of the permutation key.

4. Use the next 8-bits to rotate the rows to the left.

5. Permute the columns again using the last 4 bits of the permutation key.

6. The output of first iteration is XORed with the next iteration key.

7. Repeat the steps 1-6 to yield an efficient cipher-text.

Decryption is simply the reverse of encryption procedure by using reverse S-box and inverted lookup table.

V. COMPARISON OF PERFORMANCE

This section presents a comparative study between the Twofish algorithm and the Lightweight algorithm. It explains the experiment in order to evaluate their performance to establish the best algorithm between the two, suitable for encrypting the data in smartphone devices. It then reports the results and discusses them.

Performance of the encryption algorithms is evaluated considering the following parameters.

a. Battery power consumption
b. CPU frequency
c. Time of execution

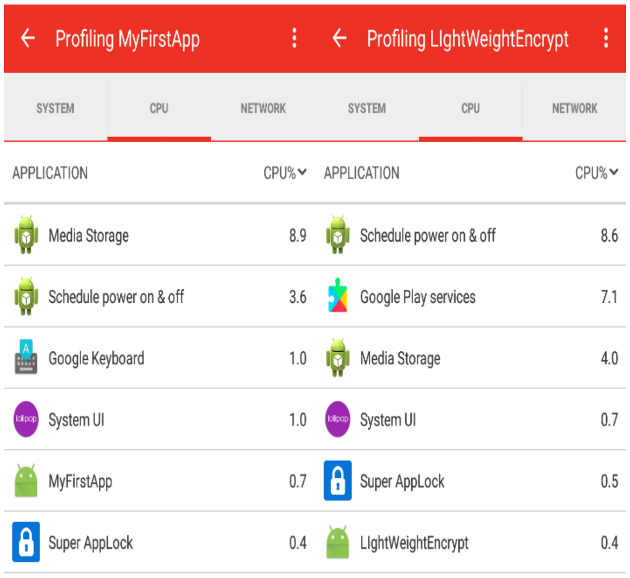
Table 2 shows the related contrast and analogy between both the encryption algorithms.

Table 2.Comparitive analysis

Parameter	TwoFish	Lightweight
Type	Symmetric	Symmetric
Input Block Size	128	128
Output Cipher Text	128	128
Original key Size	128,196,256	1008
X-ored key Size	128	128
S-box Substitution	Yes	Yes
Cipher Block Chaining	Yes	Yes
Round Key generation	Yes	No
Row Rotation	No	Yes
Column Permutation	No	Yes

VI. EXPERIMENTATION AND COMPARISON OF PERFORMANCE

A small sized data is used for the experimentation of two different encrypting algorithms (Twofish and lightweight). The comparative experimental results for encrypting and decrypting respective files are shown in figure 7. The Twofish and Lightweight algorithms are tested on same data using certain parameters. By meticulously analyzing the graph below we observe that Twofish has more memory usage as compared to lightweight. Variation in CPU frequency is observed. The execution time taken by Twofish is more than that of the lightweight algorithm.



Profiling MyFirstApp

SYSTEM | CPU | NETWORK

APPLICATION	CPU%
Media Storage	8.9
Schedule power on & off	3.6
Google Keyboard	1.0
System UI	1.0
MyFirstApp	0.7
Super AppLock	0.4

Profiling LightWeightEncrypt

SYSTEM | CPU | NETWORK

APPLICATION	CPU%
Schedule power on & off	8.6
Google Play services	7.1
Media Storage	4.0
System UI	0.7
Super AppLock	0.5
LightWeightEncrypt	0.4

Figure7. CPU percentage of each application

By meticulously analyzing the table above we can notice that Twofish has more percentage of CPU usage as compared to Lightweight

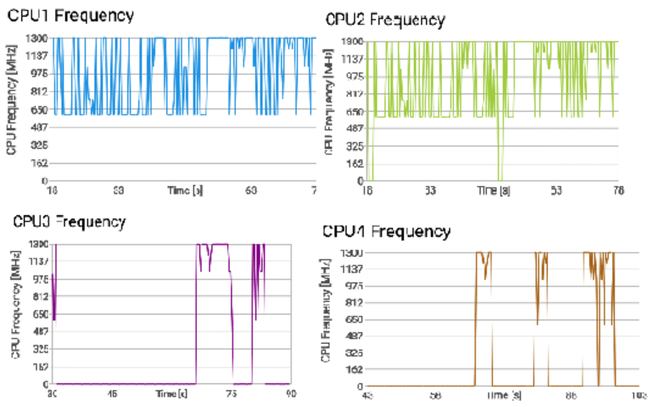


Figure 8.Twofish encryption CPU load

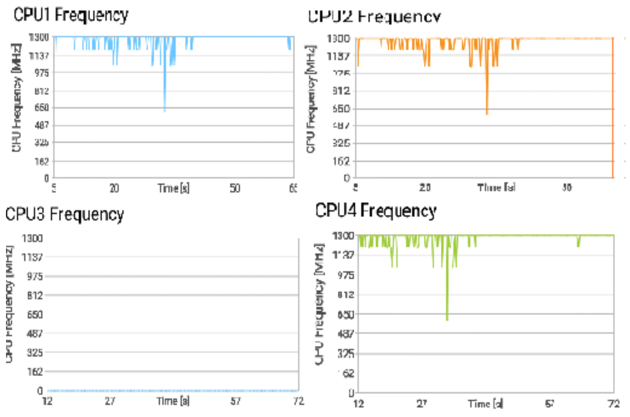


Figure 9. Lightweight encryption CPU frequency

The frequency of CPU at the beginning of Twofish algorithm is greater than that of Lightweight algorithm

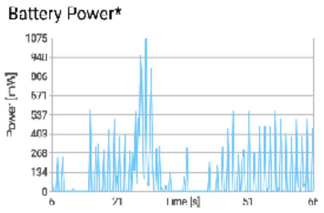


Figure 10.a.Twofish

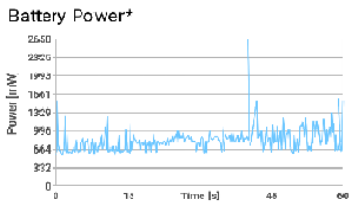


Figure 10.b. Lightweight

There is a large variation in battery power consumption of Twofish algorithm with respect to lightweight. It reaches the maximum peak for a certain time period, and drops down to zero. Whereas in lightweight, the scale starts from 644 mw, thus reducing power consumption

VII. CONCLUSION

The existing work on traditional and modern encryption algorithm has been surveyed in this paper. These encryption algorithms are studied and analyzed well so as to promote the modern encryption techniques and also to ensure the security proceedings. New encryption techniques are evolving every day and hence faster and more secure encryption techniques will always work promising a higher security rate. The primary goal is to make a comparative analysis of two android based encryption algorithm. It is clear from the experimentation that for smartphone devices Lightweight encryption algorithm is more efficient.

ACKNOWLEDGMENT

This work was supported by Department of Computer Engineering of Vidyavardhini's College of Engineering and Technology

REFERENCES

[1] Dr. S.A.M Rizvi, Dr. Syed Zeeshan Hussain, Neeta Wadhwa, "Performance Analysis of AES and TwoFish Encryption Schemes", International Conference on Communication Systems and Network Technologies, 2011.

[2] William Zegers, "A Lightweight Encryption and Secure Protocol for Smartphone Cloud", 2015 IEEE Symposium on Service-Oriented System Engineering.

[3] Laukendra Singh, Rahul Johari, "Comparative Analysis of Cryptography Cipher Techniques", International Conference of Advance Research and Innovation (ICARI-2015.

[4] Shun-Lung Su, Lih-Chyau Wuu, and Jhih-Wei Jhang, "A New 256-bits Block Cipher –Twofish 256", Computer Engineering & Systems, International Conference in IEEE, 2010, pg 166 – 171.

[5] B. Kaliski, "Password-Based Cryptography Specification Version 2.0", RFC 2898, September 2000.

[6] Joan Daemen and Vincent Rijmen, "The Design of Rijndael, AES - The Advanced Encryption Standard", Springer-Verlag 2002.

[7] Alber O. Montoya B.*, Mario A. Muñoz G.† and Sergio T. Kofuji, "Performance Analysis of Encryption Algorithms on Mobile Devices".

[8] J. Kilian and P. Rogaway, "How to Protect DES Against Exhaustive Key Search," Advances in Cryptology — CRYPTO '96 Proceedings, Springer-Verlag, 1996, pp.252–267.