An Optimized Cryptography Algorithm And Key Exchange Method for Small Scale Devices

Maulik Patel
Wireless and Mobile Computing(Master of Engineering),
GTU PG School,Gujarat Technological University,
Ahmedabad,India.
maulikpatel3110@gmail.com

Mr. I Prabu
Senior Technical Officer,
Spatial Sciences & Disaster Management Group, Centre for Development of Advanced Computing, Pune, India
iprabu@cdac.in

Abstract - Geo Informatics team will port its current method of surveying which is on paper based on web based and mobile application. All this survey data are confidential for Government purpose. With the current revolution of smart devices, this paper exercise can be converted to digital experience, thereby reducing, analyzing time, increasing accuracy and this data should be secure. These systems require efficient and fast encryption algorithm because all data are confidential data and the data should not to be exposed. Various algorithms are already existing. As our system requirement symmetric algorithm is applicable. A number of symmetric key encryption algorithms like Vernam cipher, DES, 2 DES, AES, IDEA provide better security, but their execution speed is slow. These algorithms have their own security strength, execution speed and efficiency. The cryptography algorithm should be lightweight, fast execution and highly secure. In this study, the new security algorithm is proposed based on exiting symmetric key algorithm. This algorithm efficiency and execution speed will be better than other algorithms and security will be quite unbreakable. Integrity, confidentiality, authenticity, non-repudiation of data will also be ensured.

Keywords- encryption time , execution speed , encryption , decryption, security,secure fast cryptography algorithm.

*****

I. INTRODUCTION

With the current revolution of smart devices, paper based exercises can be converted to digital experience, thereby reducing, analyzing time, increasing accuracy and providing online monitoring of field survey data. This field survey data are confidential and highly secure. Survey data are in various forms, like images, video, text based data and location data. Such high security data would not be stored and transfer over the Internet as plain data. The security algorithms available can secure data, but the problem is the speed of securing data for real time applications like the field survey. Currently available security algorithms are relatively slow in execution even on a high speed computer. In this study, security algorithm needs to be built for relatively low end digital smart devices like android mobile phones / tablets. These devices have less processing capability. Here the new security algorithm is proposed to make fast and secure data encryption with the new key exchange method. The new cryptography algorithm is based on elementary operation like Arithmetic, Rotation and bit-wise XOR (ARX In this article, MSEA is proposed to provide flexibility to the user according to his needs of security level with more secure and fast implementation. The SFCA is a symmetric block cipher algorithm. This algorithm works with different size of plain text data and key size.). Symmetric algorithms have problem of secret key exchange problem. In this study, new key exchange method based on a unique ID number of devices like International Mobile Equipment Identity (IMEI number). This new cryptography algorithm is working efficiently with the new key exchange method.

Cryptography algorithm is of two types Symmetric and Asymmetric algorithm. It can further classify these algorithms into block cipher, stream cipher, and public -private key and secret key. In this study main focus on symmetric key algorithm. Various types of symmetric key algorithms are DES, 3DES, AES, RC2, RC4, RC5, IDEA and BlowFish.

Here a description about some algorithms are given

DES: Data Encryption Standard was the first encryption standard to be recommended by NIST. It takes 64 bit plaintext and key of 56 bits as an input and it produces a 64 bit output. It's made up of S and P boxes functions. P-boxes transpose bits and S-boxes substitute bits to generate a cipher. Many attacks and its methods already recorded DES weaknesses, so it insecure at some level. [1] [4]

3DES: is an enhancement of DES. It works with 64 bit block key size of 192 bits. It's like the DES but applied three times, that's why its safe time is also increased. It is slower than other block cipher algorithm. [1]

RC2: 64-bit block cipher and key size is variable. The range is from 8 to 128 bits. RC2 is vulnerable to attack by a related key attack using 2^34 chosen plaintexts. [1]

Blowfish**:** Block cipher 64-bit block. It can be used as a replacement for the DES algorithm. It uses variable length key, and it's rang is from 32 bits to 448 bits, 128 bits is default. Blowfish is license-free unpatented. It has variants of 14 rounds or less. Blowfish is successor to TwoFish. Blow fish is stronger than DES, but it susceptible against reflectively weak keys. [4]

AES: it is a block cipher. Length of the key is variable 128, 192, or 256 bits; 256 is default. It encrypts data blocks of 128 bits. Its round is 10, 12 and 14 depends on the key size. Encryption is fast and flexible. AES has been carefully tested for many security applications and it is complex. [1]

RC6: It is a block cipher and it's derived from RC5. It has a block size of 128 bits and it supports key sizes of 128, 192 and 256 bits. It is considered as Advanced Encryption Standard by some reference. [1]

As per the study, any algorithm security, strength and Vulnerabilities depend on this much parameters: Key Size, Key type (Public-Private key / Secret key) Block cipher / stream cipher, Structure of the algorithm, Nature (Closed / Open), Input Data size, Input Data type (image, text), number of rounds and complexity of algorithms.[1]

If we increase the key size of algorithm security is also increased. Blowfish algorithm works well with text data type than the image data type. [4] Algorithm strength is depend on the structure on which it works like DES algorithm uses the balanced feistel network, AES uses Substitution-Permutation network, RC2 uses sources heavy Feistel network and BlowFish uses feistel network. [6]

In this Research, a new Symmetric block cipher algorithm named SFCA (Secure, Fast Cryptography algorithm) is proposed. It uses the concept of already existing algorithms to make new which can run on low memory and low processor devices efficiently and fast. It is based on ARX cryptographic design technique. It is simple in nature due to the use of combinations of elementary operations like modular addition, bit-wise XOR, shifting operation. In this algorithm plain text block size is variable and the secret key size depends on plain text block. In this, the number of encryption round is 4 and total 16 sub key is generated in 16 rounds, each round uses the 4 sub key. The key feature is here, we can use any size of the Input Plain text block as per our requirement. Key generation, Encryption/Decryption is significantly efficient and fast and it can run on small memory and low processor like Digital devices like mobile phone efficiently.

A. SFCA algorithm

It is a Symmetric block Cipher algorithm. It uses CBC (Cipher Block Chaining) mode among various modes of operation as its best suitable. It works on any size of Plain text (64,128,256 bits or more). Key size is dependent on Input block size and it is half of block size. In key generation method, 16 sub-keys are generated in 16 rounds. Encryption algorithm works in 4 rounds. Structure of encryption round is like a Festal structure. There is a function name M function (Mixing function) which is applied on right half of the plain text block. Each round uses 4 sub keys for encryption of data and sub keys are used in M function. The output of one, round which is cipher block, it is applied as input on the next block for encryption.

In this cryptographic algorithm, key size is depend on the input plain text size. As per this algorithm, key size is half of input block size. If input block size is N then key size required for its encryption is N/2. If input block size are 64, 80,112,128,156,192,256 bits, then key size are respect to 32, 40, 56, 64, 78, 96, 128 bit. Here we consider 256 bit as an Input plain text block and key size is 128 bits for example purpose. In this algorithm, Main key is generated randomly.

TABLE I.

TABLE II. KEY SIZE RESPECT TO INPUT DATA SIZE

Input Block Size (N bits)	Key size(N/2 bits)
64	32
128	64
192	96
256	128

In this Cryptographic algorithm total 16 sub keys arerequired. Sub keys are generated from Main key K. Subkeys are generated in following 3 steps.

1) Initial Permutation
2) Key Compression
3) Sub key generation

1 Initial permutation for key generation

Initial permutation is performed on Main key which is made from IMEISV number. In this step all bits are randomly permuted using IP tables. So after this step new 128 bit key stream will be generated.

2 Key compression

In this step main key compressed to half of its size and it's used for further key generation. Here we consider main key size as 128 bits. In this Key Compression step 128 bits are converted to 64 bits. In this step 128 by are divided into 4 blocks. After division each block size will be of 32 bits. Here we consider 128 bit key stream is divided into 4 blocks P, Q, R and S .Each block size is 32bits.

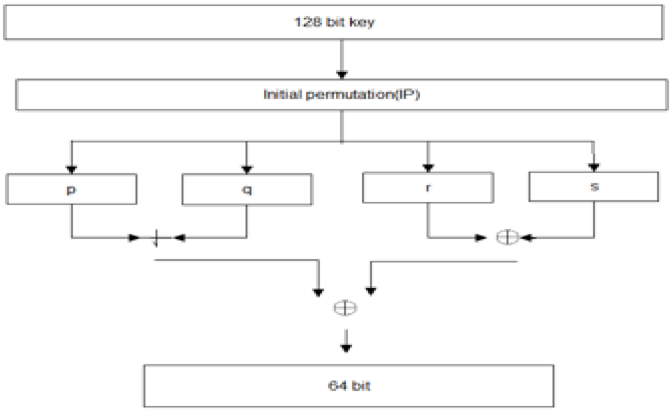


Figure 1: Memory map for background privileged access only

Operation performed onto the key stream to generate 64bit output is explained below.

$P = P + Q$

$R = R \oplus S$

$P = P \oplus R$

After this step output is compressed 64 bits. This 64 bit key stream is used as input for key generation step as follows.

2 Key generation

In key generation, total 16 sub keys are generated. As per following figure 16 sub keys are generated.

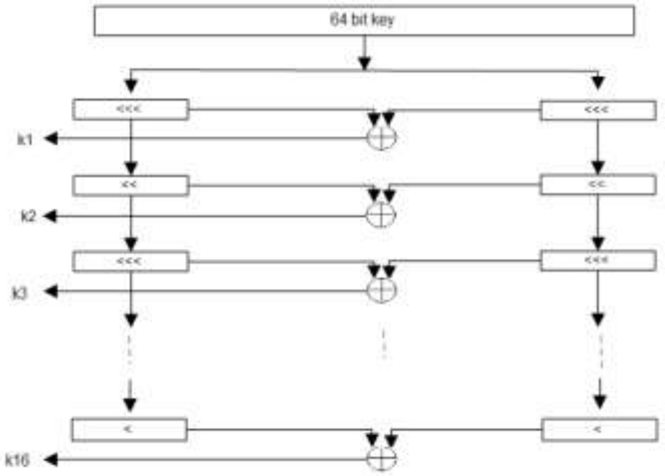


Figure 1: Key generation rounds

In this key generation step at each step different left shift operation is performed. In 1,3,4,7,9,13 and 15th round 3 left shift operation is performed. In 2,6,8,10,12 and 14th round 2 left shift operation is performed. In 5, 11 and 16th round only 1 left shift operation is performed.

TABLE III. LEFT SHIFT OPERATION RESPECT TO ROUND NUMBER

Round no	Left shift operation
1,3,4,7,9,13,15	3 shift
2,6,8,10,12	2 shift
5,11,16	1 shift

In this step 64 bits are divided into two halves left and right. Each half is about 32 bits left half (LH) and the right half (RH). In 1st step 3 left shift operation is performed on both halves. The output of both halves XORed It generate sub key k1. After that 2 left shift operation is performed on previous block and then output of it XORed which generate sub key K2. And likewise all 16 sub keys are generated. At different round different left shift operation is performed. One step output used as input of the next step. By this step sub keys are k1, k2, k3… k16 are generated.

b Encryption process

The Initial permutation operation is performed on the original plain text block. IP used to rearrange all the bits of plain text. IP used key generation step is different in this step.

1 Initial permutation for encryption process

The initial permutation operation is performed on the original plain text block. IP used to rearrange all the bits of plain text. IP used key generation step is different in this step.

2 Encryption rounds

During encryption rounds output of IP step is divided into two left and right halves (LH and RH). On the left halve Initial permutation is performed. The M function operation is performed onto right halve (RH). M function required 4 sub keys (k1, k2, k3 and k4). Output of M function is XORed with an output of IP of left halve (LH).Now output of XOR operation which is at left side it become right halve and right halve become left halve. As per the rules, in next step again M function performed onto right halve.

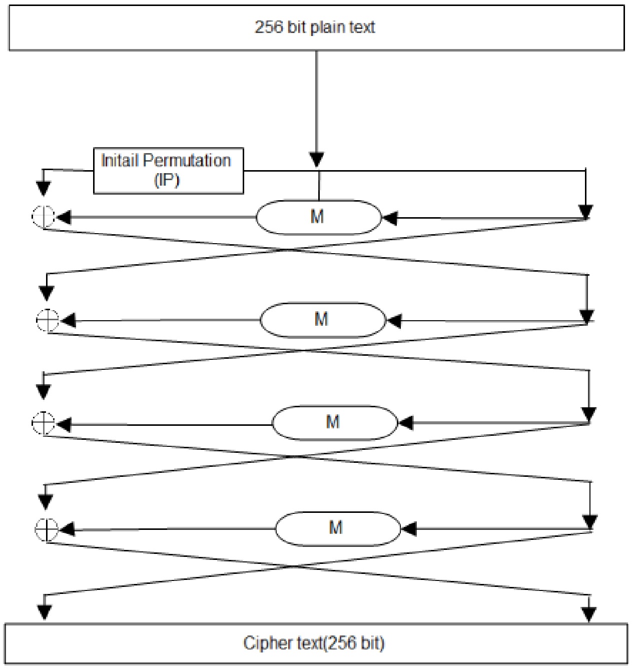


Figure 3: Key generation rounds

In next step again M function performed onto right halve. This time M function required next 4 sub keys (k5, k6, k7and k8). Now Output of M function XORed with left halve and likewise all four will execute. At the end of operation 256 bits cipher text will be generated.
This cipher block also goes as input of next plain text block for further operation as requirement of CBC mode.

3 M function(Mixinf function)

M function is main operation for encryption step in this algorithm. It is called mixing function. It mix the all the bits so that to attack on the cipher text to get plain text it become difficult. In M function , right part of previous step is dived into 4 part. Here named as A ,B,C and D. Each half is of 32bits. On this block different operation is performed.
In this step as per figure operation is perforemd as per following steps

$G = C \oplus D$
$H = G \oplus A$
$E = A \oplus B$
$F = B \oplus C$

After this step another operation of M Function is performed on this varibale. Each time M function required 4 sub keys for encryption.
As per following method operation will execute
$T = E + k1$
$U = T \oplus F + k2$
$V = U \oplus G + K3$
$W = V \oplus H + K4$

Now another operation is perforemed onto the T,U,V and W . Two operation is performed on the T and V . First is modular addition operation is performed on T and V which generate output as TV. XOR operation performed onto TV nd V which generate the output as VT. These TV and VT are concated into X variable. Like wise these both operation are performed onto U and W which generate the Y.

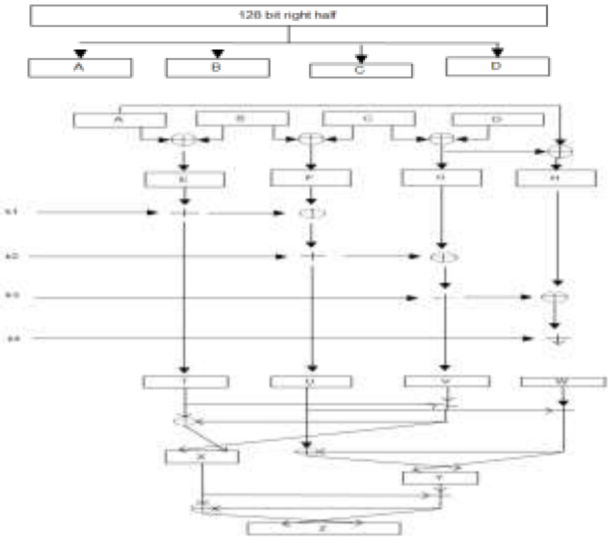


Figure 4: M function (Mixing function)

Now another operation is perforemed onto the T,U,V and W . Two operation is performed on the T and V . First is modular addition operation is performed on T and V which generate output as TV. XOR operation performed onto TV nd V which generate the output as VT. These TV and VT are concated into X variable. Like wise these both operation are performed onto U and W which generate the Y.

$TV = T + V$
$VT = TV \oplus T$
$X = VT$ concate TV

$UW = U + W$
$WU = UW \oplus U$
$Y = WU$ concate UW

$XY = X + Y$
$YX = XY \oplus X$
$Z = YX$ concate XY

The Same method is applied on the X and Y to make its final out put as Z which is of 128 bit. Which we called output of M function. This whole procresses is repeated for all M function. This output goes to as input to XORed operation with left halve of encryption round

This optimized cryptography algorithm will be implemented using java language.

B. Key exchange method

Symmetric key cryptography algorithm required a sharing of secret key to either side. Sharing a secret key in secure manner is a main problem in Symmetric key cryptography algorithm. It is necessary to send key by which data was encrypt to receiver side for decryption of that data.

There are many way by we can exchange key between sender and receiver side. RSA algorithm used to exchange the key by encrypt the secret key by public key of receiver. But RSA algorithm is not efficient to use for exchange of key each time as our aim is to reduce execution speed. Another way for key exchange is Diffie-Hellman key exchange algorithm. But it is mainly used with peer to peer communication.

Here I proposed a new concept of sharing a key for decryption applicable in mobile device or any devices which contain global unique id.

Every mobile device has a unique IMEI (international mobile equipment identity) number and IMEISV (international mobile equipment identity Software version) number. IMEI number is of 15 character and whereas IMEISV number is 16 character or 128 bit. So here we can get benefit of IMEISV number as unique key.

We can use IMEISV number as main key Km. Km is used for further step and key generation. Km is used for encryption of data. For decryption of data Km need to be send another side. But we cannot send Km in plain text for security purpose. I proposed a new key exchange concept for this type of application. This system is not applicable to all open user system. It is only applicable to system which have limited user with purpose with central system. When user uses the first time system its IMEISV number is registered. Registration process happen only first time. After successfully joining to system we can send data in encrypted form using secret key. At the receiver side same functionality will be there and it convert key IMEISV number to Main key. This main key is used as regular decryption operation. All this shown in fig below.

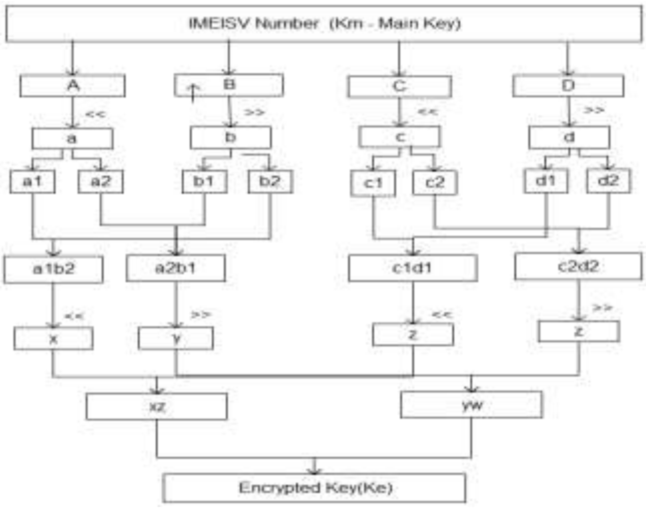


Figure 5: Encryption process of key to make main key

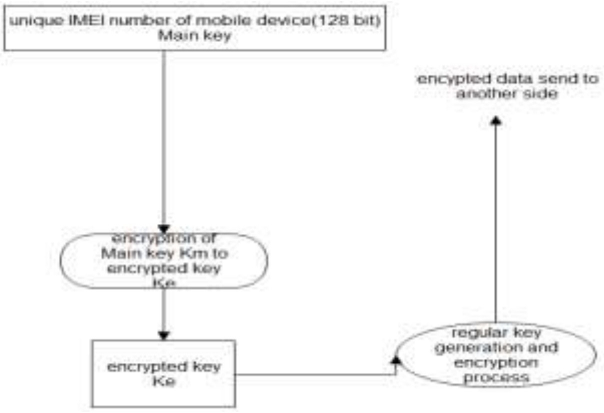

unique IMEI number of mobile device(128 bit)
Main key
encypted data send to another side
encryption of Main key Km to encrypted key Ke
encrypted key Ke
regular key generation and encryption process

Figure 6: Encryption key at client side and encryption of data

So for sharing a key we need to do processing on key for encryption. In this concept a main key Km is encrypted further without any key. For encryption of key some processing will be done on key. Some processing like rotation of key, some bit adding or removing to key like other some simple operation. This will generate the other encrypted key.

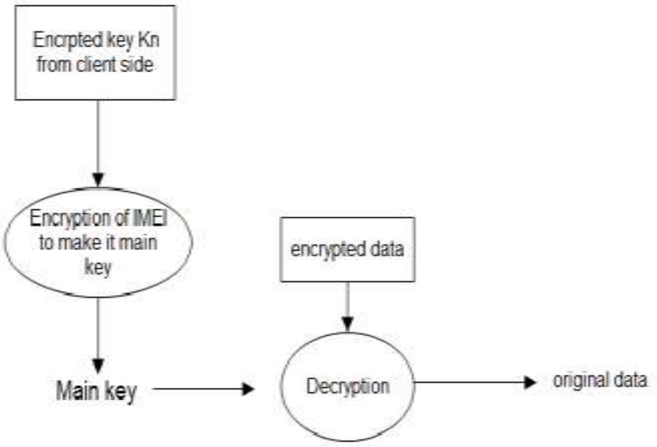

Encrpted key Kn from client side
Encryption of IMEI to make it main key
Main key
encrypted data
Decryption
original data

Figure 7: Decryption of data

At the receiver side same functionality will be there and it convert IMEISV number to encrypted key. We will get same key which was used for encryption process same key will used for decryption.

II. IMPLEMENTION AND RESULT

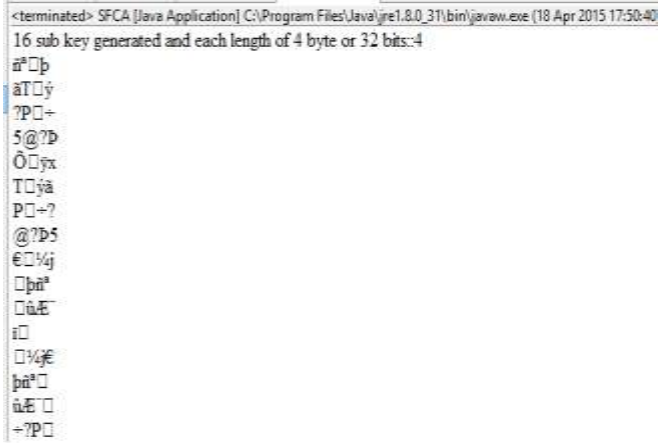
As per study new SFCA cryptography algorithm is designed. This algorithm is implemented using JAVA language in Eclipse IDE.

At current progress algorithm is designed and implemented in JAVA and encryption of sample data is done. Here it shows key generation process output, Encryption of sample data, encryption of input key for main key generation.

A. Key Generation

As per SFCA algorithm input key size is depend on input data size. Input key size is half of the input data size. Here we consider key size is 128 bit. Key Generation process generate the 16 sub key of 32 bit size each.

So, here it is shown in figure generation of 16 key from main key.



```
<terminated> SFCA [Java Application] C:\Program Files\Java\jre1.8.0_31\bin\javaw.exe (18 Apr 2015 17:50:40
16 sub key generated and each length of 4 byte or 32 bits::4
ñª□þ
ãT□ý
?P□÷
5@?Þ
Ô□ÿx
T□ýà
P□÷?
@?Þ5
€□¼j
□þñª
□ûÆ¯
i□
□¼j€
þñª□
ûÆ¯□
÷?P□
```

B. Encryption of data

The 16 sub key generated as above figure is used for encryption of data. By this algorithm different size of data can be encrypted. Here we consider input data size of 256 bit.

The sample output of encryption is shown below.

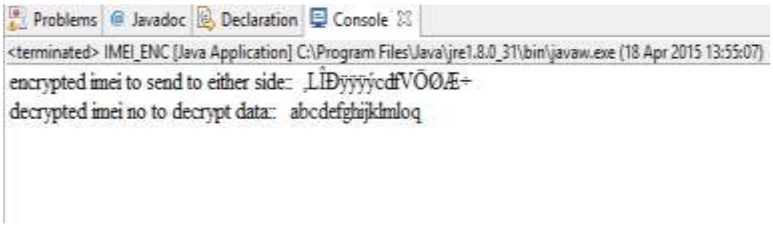
```
Original data::A Storm of Swords is the third of seven planned novels in A Song of Ice and Fire, a f
Encrypted data::
[illegible]
```

```
<terminated> SFCA (2) [Java Application] C:\Program Files\Java\jre1.8.0_31\bin\javaw.exe (25 Apr 2015 10:49:05)
Original data:The emergence of Sachin Tendulkar and Anil Kumble in 1989 and 1990 was to herald an era of Indian cricket that was dominated by stars and individual brilliance. Sachin Tendulkar became arguably the b
Encrypted data:
[illegible]
```

C. Encryption of IMEI to make it main key

As per study main key is generated from IMEISV number. We can't use IMEISV number as direct main key because of security issue.

So here some process is apply to encrypt the IMEISV number to generate the key. This encryption process only use rotating of bits, shifting and rearranging of bits in the data. Here main key is generated from input key.



```
<terminated> IMEI_ENC [Java Application] C:\Program Files\Java\jre1.8.0_31\bin\javaw.exe (18 Apr 2015 13:55:07)
encrypted imei to send to either side:: ,LÏÐÿÿÿýcdfVÕØÆ÷
decrypted imei no to decrypt data::  abcdefghijklmloq
```

III. RESULT AND ANALYSIS

In this study, the new proposed algorithm SFCA and new key exchange method is analyzed. Though SFCA can work with any size of data size but here it is analyzed with 256 bit block size and 128 bit input key size.In it every block is encrypted using different sub key generated from main key.

Main key is generated from IMEISV number by doing some encryption on it without key so we can secure the main key.

Here different algorithms like DES,,AES and SFCA algorithm is analyzed under the Intel® Core™2Duo CPU T6570 @ 2.10GHz with 3GB RAM. Algorithms are implemented in eclipse Juno.

SFCA is simple in nature and also it is secure from timing attacks because of ARX design pattern. Circular rotation and addition provides strong diffusion by creating the non linearity in encrypted message.

SFCA provides strong diffusion. According to result encrypted data contain no repeated bits or data. Differential cryptanalyst attack can not break the the cipher text. Its too tough to get origianl data back from cipher text using diffirential cryptanalyst and timing attack.

SFCA algorithms have a capability to take variable size of data.As we consider input data size is 256 bit block and as per algorithm 128 bit key is required to encrypt it. If we consider brute-force attack to break it. Its nearly impossible to break it.

Here comparison of various algorithm DES, AES and 3 DES is given with the parameter of execution speed and input data size.

In this below table Average execution speed for various input data size 5, 15, 25,35,45,55 and 65 is given. Input data size is in kilo byte and time is in second.

TABLE IV. COMPARISION OF ALGORITHMS

Input File Size(KB)	Encryption Execution Time(mille seconds)		
	DES	AES	SFCA
5	219	321	132
15	592	783	330
25	893	1230	477
35	1213	1834	673
45	1547	2235	817
55	1870	2787	1020
65	2165	3108	1186

In Below figure graph is shown for above table data. Here it says that SFCA algorithm have high execution speed than DES and AES.

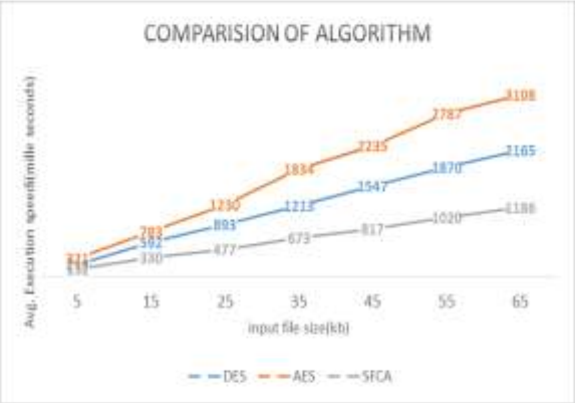


Figure 9: Comparison of algorithms

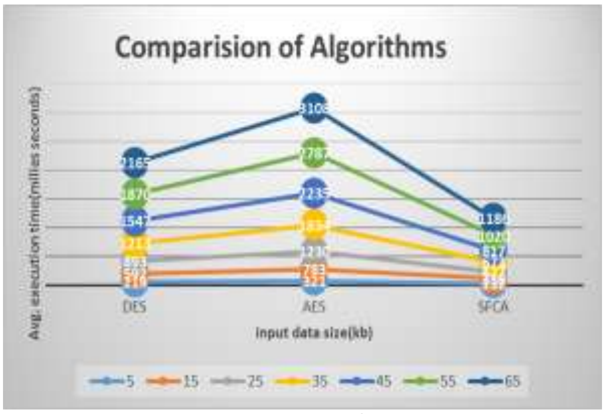


Figure 8: Comparison of algorithms

In this below table Average execution speed for various input data sizes are given. Input data size is in Megabyte and time is in Mille second.

TABLE V. COMPARISION OF ALGORITHMS

Input Data Size(MB)	Encryption Execution Time(seconds)		
	DES	AES	SFCA
1	32.711	44.234	17.242
2	59.774	84.783	30.314
2.91	93.135	135.351	47.651
3.84	113.40	163.239	63.502
5.50	178.203	249.375	90.918

In Below figure graph is shown for above table data. Here it says that SFCA algorithm have high execution speed than DES and AES.

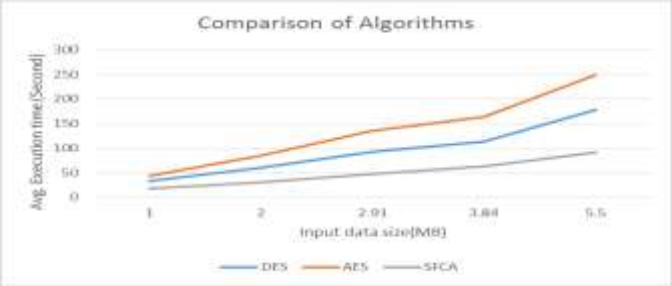


Figure 10: Comparison of algorithms

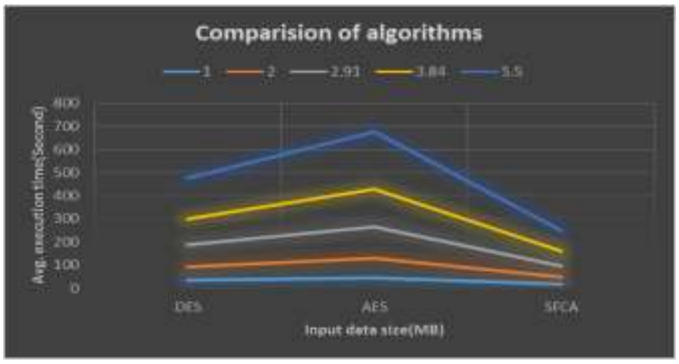


Figure 11: Comparison of algorithms

So from this experiment on various data size, we can say that SFCA algorithm have better execution speed than other algorithm. Proposed algorithm have good improvement over other algorithm.

IV. CONCLUSION

In this study I designed an algorithm which execution speed is better than existing algorithms with no compromise with security constraint. As I designed this algorithm using an ARX operations like Arithmetic operation like modular addition, bit wise operation XOR and rotation of bits. It is made using less complex operation which can execute faster in low processor and low memory devices. This algorithm is also reversible so decryption time is also same as encryption time. This algorithm also working with Input data type as images, text and media in same manner unlike Blowfish and DES. New key exchange method based on IMEISV number is work efficiently with this SFCA.SFCA algorithm speed is execute faster than DES, AES, Blowfish and IDEA. SFCA work efficiently with small scale devices with low processor and low memory devices.

V. REFERENCES

[1] Yogesh Kumar, Rajiv Munjal, Harsh Sharma **“**Comparison of Symmetric and Asymmetric Cryptography with Existing Vulnerabilities and Countermeasures” IJCSMS International Journal of Computer Science and Management Studies, Vol. 11, Issue 03, Oct 2011

[2] Sombir Singh, Sunil K Maakar and Dr. Sudesh Kumar “A Performance Analysis of DES and RSA Cryptography” IJETTCS - Volume 2, Issue 3, May – June 2013

[3] K.Brindha, Ritika Sharma, Sapanna Saini”Use of Symmetric Algorithm for Image Encryption”, International Journal of Innovative Research in Computer and Communication Engineering Vol. 2, Issue 5, May 2014

[4] Dr. J. Abdul Jaleel, Jisha Mary Thomas “Guarding Images using a Symmetric key Cryptographic Technique: Blowfish Algorithm “

[5] Lanxiang Chen, Shuming Zhou “The Comparisons between Public key and Symmetric key Cryptography in Protecting Storage Systems.” ICCASM 2010

[6] Shadi R. Masadeh, Shadi Aljawarneh, Nedal Turab “A Comparison of Data Encryption Algorithms with the Proposed Algorithm: Wireless Security”. Faculty of Information Technology Isra University

[7] Ritika Chehal, Kuldeep Singh “Efficiency and Security of Data with Symmetric Encryption Algorithms” Volume 2, Issue 8, August 2012

[8] R.Satheesh Kumar, E.Pradeep, K.Naveen, R.Gunasekaran “An Enhanced Security Algorithm for Wireless Application using RSA and Genetic Approach “–IEEE PAPER 2011

[9] Comparison of Symmetric and Asymmetric Cryptography with Existing Vulnerabilities and Countermeasures-IJSAM JOURNAL

[10] An Empirical Study to Compare the Performance of some Symmetric and Asymmetric Ciphers, IJSAM JOURANAL-2012

[11] A. Nadeem, “A performance comparison of data encryption algorithms,” IEEE Information and Communication Technologies, pp. 84-89, 2009.

[12] Dr. Najib A. Kofahi “An Empirical Study to Compare the Performance of some Symmetric and Asymmetric Ciphers”, International Journal of Security and Its Applications Vol.7, No.5 (2013), pp.1-16

[13] Ritika chehal, Kuldeep singh.” Efficiency and Security of Data with Symmetric Encryption Algorithms.” International Journal of Advanced Research in Computer Science and Software Engineering. Volume 2, Issue 8, August 2012

[14]R.Satheesh Kumar, E.Pradeep, K.Naveen, R.Gunasekaran”ENHANCED COST EFFECTIVE SYMMETRIC KEY ALGORITHM FOR SMALL AMOUNT OF DATA” 2010 International Conference on Signal Acquisition and Processing.

[15] Joel James, “New Symmetric key Cryptographic algorithm using combined bit manipulation and MSA encryption algorithm: NJJSAA symmetric key algorithm” 2011 International Conference on Communication Systems and Network Technologies