A Sturdy Compression based Cryptography Algorithm using Self-Key (ASCCA)

K Lakshmana rao [#1], Hima Bindu Maringanti [*2], Balajee Maram [#3]

[#] CSE-Department, GMRIT, Rajam, 532127, India, [*]Department of Computer Applications, North Orissa University, Baripada,757003, Mayurbhanj, Odisha, India, [#] Ph.D (CS) Research Scholar, Research and Development Centre, Bharathiar University, Coimbatore.

[1] lakshmanarao.k@gmrit.org

[2] profhbnou2012@gmail.com

[3] balajee.journal@outlook.com

***Abstract*— Storage space is very precious in hand-held devices. Thus it is very important to utilize the space in phone's Internal and external memory effectively. This paper proposes a technique for compression of data and security of compressed data in hand-held devices. The proposed technique is based on SMART DICTIONARY BASED ENCODING and DECODING (SDBED) which compresses the data in phone's memory. When hand-held device receives a file/SMS, long English words will be replaced by small integer numbers. Then these lengthy English words are compressed. Thus in different phases most of the English words are compressed and the compressed text is self-encrypted with self-key. This paper describes the SMART DICTIONARY BASED ENCODING and DECING (SDBED) Compression Technique for files/SMS in hand-held devices.**

Keyword- Dictionary, Compression, SDBED, DRDP, SMS.

I. INTRODUCTION

In Mobile communication, one of the important items in communication technology is SMS. SMS has now become a popular and uses by every individual. The main motto of the current technology is to use the resources in a more convenient, easy and cost-effective way. The mobile phones and smart phones have to follow best technology for text compression and it should save the memory space. The current technology admits up to 160 characters for a single SMS.

In this paper, we propose an algorithmic technique of compressing and encrypting SMS text for mobile phones. The aim of this proposed algorithm is to establish a cost effective and lossless compression algorithm suitable for handheld devices like mobile phones which are having very less memory with low processing speed. The SMART DICTIONARY BASED ENCODING and DECODING (SDBED) is a technique to compress small messages, which requires optimal space, consumes less time, low overhead. Finally SMART DICTIONARY BASED ENCODING and DECODING (SDBED) reduce the communication costs. SDBED is used to design a temporary dictionary for compression.

In this Research Paper, ASCCA is proposed the alternative technique for Symmetric and Asymmetric Cryptography algorithms. ASCCA have the following characteristics:

- It creates a temporary dictionary and no need to send along with file/SMS.
- It is a symmetric block cipher.
- The Architecture of the ASCCA is very simple that uses bitwise XOR, DRDP Converting method, Fibonacci & Lucas Number Series.
- For each and every Plain-Text 256-bit Block, it generates Sub-Keys and Final-Keys. It is one of the important features of ASCCA.
- The generation of Final-Key for each Input-Block is depends on Fibonacci and Lucas Number Series.

This Paper is organized as follows: Literature Survey is covered in Section 2. Proposed algorithm, which includes Dynamic Dictionary Creation, Word-Compression, generation of Round-Key and Final-Key for each Round and the encryption/decryption schemes, is covered in section 3. ASCCA Encryption/Decryption process analysis is covered in Section 4. Security Level of ASCCA Encryption/Decryption process is covered in Section 5. ASCCA Encryption/Decryption algorithm illustration is covered in Section 6. Finally, the conclusion is covered in Section 7.

II. LITERATURE SURVEY

2.1 Data communication

In Data Communication, there is a need of a sender, a recipient, message, and communication network. Here the intention of the sender is to send data to the recipient. If the recipient is in online, then he will receive the message instantly. Otherwise the message will be stored in Message-Queue/Inbox. The Communication process is successful when both the persons exchange their information successfully [1]. Communicating with others involves three primary steps:

- Thought: Some item/feeling comes in the mind of the sender. This can be a concept, idea, information, or feelings.
- Encoding/Encryption: A message is sent to a recipient in words or other symbols.
- Decoding/Decryption: When the recipient translates the words, symbols or cipher into the information that a person can understand.

On the other hand secure Data Communication needs to apply Cryptographic algorithms. But all algorithms are based on either private-key or public-key.

Data compression offers an attractive approach for reducing communication costs by using available bandwidth effectively. The Compression algorithms eliminate the repetition in data representation to decrease the size of the data. The following sections give a glimpse of the most recent research developments on text compression issues for mobile devices.

2.2 Compression

The algorithm Indexed Reversible Transformation (IRT) is proposed by Stefan Böttcher, Alexander Bültmann, Rita Hartel. The algorithm is an extended version of Burrows-Wheeler-Transformation (BWT). The BWT algorithm is a combination of run length encoding (RLE) and wavelet trees (WT). The BWT is based on position and updating substrings of compressed texts. It leads IRT is useful for a large number of applications and it saves data access time as well as space consumption in main memory. The list of existing compression algorithms and encryption/decryption algorithms are as follows:

-- Howard Cheng, Xiaobo li [2] performed compression using Quadtree compression Algorithm, during compression process partial encryption is applied.

-- Ebru Celikel, Mehmet Emin Dalkilic [3] performed experiments on a secure compression algorithm. Results are compared using different compression techniques like Arithmetic coding, Huffman Coding, Lempel-Ziv, Prediction by Partial Matching and Burrows Wheeler. Encryption is performed using symmetric key BBS PRNG. The authors applied algorithm on text file in English and Turkish.

--Masanori Ito et al. [4] proposed a method combining encryption and compression based on Independent Component Analysis (ICA) and Discrete Cosine Transform (DCT)..

--Goh Han Keat et al. [5] observed Embedded Zerotree Wavelet (EZW) encoder which specially designs for wavelet compression. Stream ciphers RC4is selected as the encryption algorithm.

--N. V. Thakur and O. G. Kakde [6] proposed the compression and encryption based on the fractal coding and spiral architecture but the compression method are lossy.

2.3 Cryptography algorithms

In this modern communication technology, whenever any sensitive data transmitted over a public channel, the hacker/attacker could get the message. So there is a need to give protection to the sensitive data which is transmitted through Internet. For protecting these transmissions, the old and modern cryptographic methods are very useful.

In every cryptographic algorithm, there is a need of either shared secret-key or private/public key pair. The strength of the encryption is based on the length of the key. When the length of the key is very small then easy to crack the key as well as decrypt the data. The required keys should be shared in symmetric cryptography and the public key should be published in asymmetric cryptography techniques.

In Symmetric Cryptographic Algorithms, a single key is shared between sender and recipient. In most of the Symmetric Cryptographic Algorithms (DES, 3-DES, AES, IDEA, RC4 and RC5) the sizes of the key are 56-bit, 128-bit or 256-bit only. In public key cryptography, sender uses public key of receiver, known to everyone, to encrypt the message and receiver uses his private key, known only to him, to decrypt the message. RSA is the one of the most famous public key algorithm which is based on Diffie-Hellman Key Exchange [1]. Most of the keys are crack able through the one of the following techniques:

- Brute-Force Attack
- Differential Cryptanalysis
- Linear Cryptanalysis

---Prakash Kuppuswamy, Dr.Saeed Q Y Al-Khalidi, have proposed an algorithm based on Modulo 37[7]. This algorithm uses two keys: k1=positive number, k2=negative number, find the inverse of both using modulo 37, giving k1', k2'.

--- A Symmetric Key Cryptographic Algorithm by Ayushi, 2010 [8]. It Generate the ASCII value of the letter and corresponding binary value and reversing the binary. There is no standard key generation method.

--- An Efficient Developed New Symmetric Key Cryptography Algorithm For Information Security By Suyash Verma, Rajnish Choubey, Roopali Soni, July 2012[9]. It is a block cipher that divides data into blocks of equal length and then encrypts each block using a special mathematical set of functions known as key.

--- A Modified Approach For Symmetric Key Cryptography Based On Blowfish Algorithm By Monika Agarwal, Pradeep Mishra [10]. It is a 64 bit block cipher with a variable key length.

2.4 Fibonacci sequence

Fibonacci [11] introduced Arabic numerals to Europe. His theorem gives a sequence (the Fibonacci sequence) in which "each number is the sum of the two preceding numbers". Thus, the sequence progresses: 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597... .

2.5Lucas Numbers

Francois-Edouard-Anatole Lucas is the French mathematician, professor. Lucas studied the Fibonacci sequence and proposed Lucas sequence. Lucas Series [12] is the sequence of numbers 1, 3, 4, 7, 11, 18, 29, 47, ... given with the following formula:

$$L_n = L_{n-1} + L_{n-2}\ for\ n > 2, L_1 = 2, L_2 = 1\ for\ the\ initial\ terms\ L_1 = 1\ and\ L_3 = 3.$$

Example: LUCAS NUMBER SEQUENCE which are

1 , 3, 4, 7, 11, 18, 29, 47,76, 123, 199, 322, 521, 843, 1364, 2207, 3571,5778, 9349, 15127, 24476, 39603, 64079 (23 numbers from LUCAS3 NUMBER SERIES)

2.6 The Double-Reflecting Data Perturbation Method

The Double-Reflecting Data Perturbation Method [13] denoted by DRDP reverberates the original data by x-axis and y-axis to achieve the perturbed data for some confidential attribute. In this method, the randomization function plays a very crucial rule, and if the function is not properly chosen it May degrade the clustering quality. The distortion operation performed to the confidential attribute is as given follows:

$$0\rho_j = \rho_{Aj} + \left(\rho_{Aj-aj}\right) = 2\rho_{Aj-aj}$$

$$\text{Where } 1 \le j \le n \text{ is a confidential attribute and j is an instance of Aj.}$$

$$\text{The } \rho_{Aj}\ is\ defined\ as\ \rho_{Aj} = \left|\frac{(\max Aj + \min Aj)}{2}\right|$$

Where $\max Aj + \min Aj$ are respectively the maximum value and minimum value of attribute Aj. The 'student' relational database before and after applying DRDP is shown in the following Table I:

TABLE I. Example of DRDP on Student Data.

S.No	Roll No	Name	Marks	Distorted Marks
1	101	Rohan Raj	78	92
2	102	Shashank Raj	89	81
3	103	Prudvi Raj	92	78
4	104	Dhan Raj	82	88
5	105	Mythily Raj	80	90

III. PROPOSED SYSTEM

The most widely used data compression schemes /algorithms are based on the sequential data compressors of Lempel and Ziv. These Statistical modelling techniques produce superior compression, but are significantly slower. In this paper, we present a technique that achieves higher compression ratio as compared to that achieved by statistical modelling techniques, and at speeds comparable to those of algorithms based on Lempel and Ziv's.

The main issue in this paper is to implement a lossless and low complexity compression of short English text for low power consuming smart devices like cell phones with small memory. The proposed scheme is concerned with two parts. The first one consists of designing the semantic dictionary and the second provides a compression / decompression technique. The basic idea is to first convert the text in SMS to an intermediate pattern by pre-processing it, and then compress the output to generate compressed Text.

A strategy called Smart Dictionary Based Encoding and Decoding (SDBED) is discussed below to achieve this. It has been observed that a pre-processing of the text before compression will improve the compression efficiency. The implementation of the Proposed Cryptography Algorithm is as follows:

3.1 Encryption Process

Phase 1: Dictionary-based Compression

Step 1: Extract words from the input file one by one and check whether the word is number or not.

Step 2: If the word is a number then insert "0C (form feed)" after the number.

Step 3: If it is word, check the length of the word. Ignore if length is less than or equal to 2.

Step 4: If length of the word is greater than 2, then check the appearance of the word.

i) In first appearance, enter the word along with the counter in temporary table.
ii) If not first appearance, replace the word with corresponding counter value from the temporary table.
iii) If the word is substring of an entry in temporary table and length of the word is more than 3, replace the word with the following:

<Starting Character, Number of Character, Counter value of entry in temp. table >

Then these bits converted into an Integer value.

Step 5: Continue the same process for remaining words.

Phase 2: Compression of Alphabetical words

After Phase 1, every word appears one time in the entire text. The text containing unique words, Integer values and integers following by "0C (form feed)". In Phase 2, each word gets compression using the following proposed algorithm:

A Compression of alphabetical words

The size of ASCII character is 1-Byte i.e 8-bits. Now the proposed compression algorithm reduces the size of each ASCII character. In the proposed system, it allocates 5-bits for each lower-case English character in the following table-II:

TABLE II. Binary codes for each character

Character	Code	Character	Code	Character	Code
A	**00000**	**J**	**01001**	**S**	**10010**
B	**00001**	**K**	**01010**	**T**	**10011**
C	**00010**	**L**	**01011**	**U**	**10100**
D	**00011**	**M**	**01100**	**V**	**10101**
E	**00100**	**N**	**01101**	**W**	**10110**
F	**00101**	**O**	**01110**	**X**	**10111**
G	**00110**	**P**	**01111**	**Y**	**11000**
H	**00111**	**Q**	**10000**	**Z**	**11001**
I	**01000**	**R**	**10001**		

If the word is not in lower-case, it allocates 2 more bit for entire word. The value '01' in the extra bit indicates the word starts with Upper-case character. The value '10' in the extra bit indicates the entire word is in upper-case.

Ex:

cat → Here all characters are in lower-case. The size of "cat" is 15-bits.

0	0	0	1	0	0	0	0	0	0	1	0	0	1	1

Cat → Here the first character is in upper-case. So the values of extra 2-bits are '0 & 1'. The size of "Cat" is 17-bits.

0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	1	1

CAT→ Here all characters are in upper-case. So the values of extra 2-bits are '1 & 0'. The size of "CAT" is 17-bits.

1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	1

Using this proposed word compression algorithm, each word is compressed.

Phase 3: Encryption is performed as shown in Fig.1.

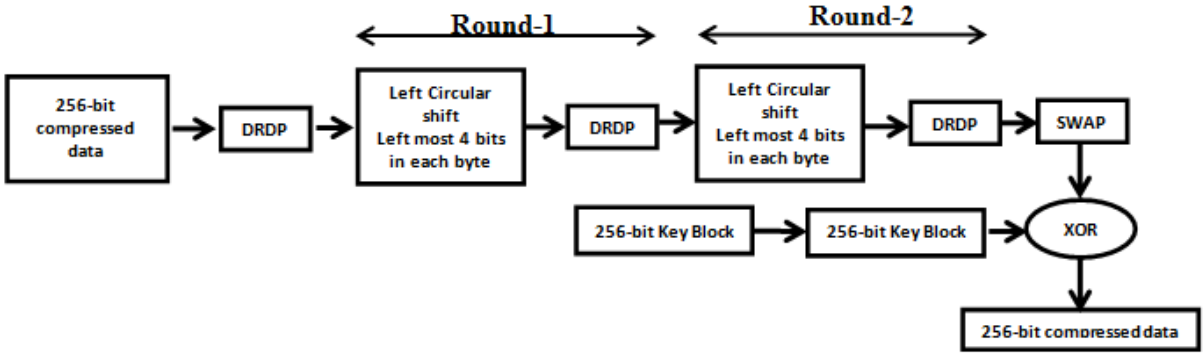


Fig. 1: Encryption process

Step 1: Convert the Compressed Data into Binary-Form.

Step 2: Divide the Binary-Data into 256-bit Blocks i.e P[1], …. , P[n]. If required the binary-data is padding with '0'.

Step 3: If total number of 256-bit chunks is even, then add one more 256-bit block which contains '0'.

Step 4: Now the binary-data have odd numbered 256-bit chunks. Assume 'n' number of 256-bit blocks. The middle 256-bit block is the key. Assume the k^{th} 256-bit block is the key.

Step 5: Divide each 256-bit chunk into 8-bit then ASCII character.

Step 6: Take 256-bit Block.

Step 7: Apply DRDP Converting Technique on *Step 6* Block

Step 8: Perform 2 Round Operations on *Step 7* Block in the following way:

- ➔ Divide 256-bit Block into 4-bit chunks. Make odd numbered chunks Left Circular Shift.
- ➔ Apply DRDP Character Converting technique.

Step 9: The Intermediate-Text is swapped as shown in the following Fig.2.

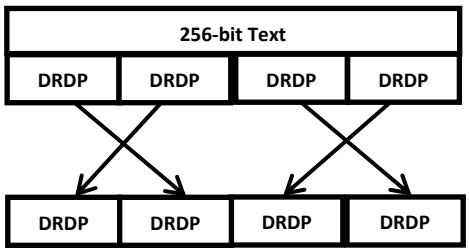


Fig. 2. Swapping of 256-bit Data

Step 10: Apply DRDP method on current 256-block and the corresponding character in key-block. So there is a chance to change in a single character in key-block.

Step 11: Apply bitwise XOR on *Step 9* Block and Final-Key, i.e. k^{th} block (Key-block which was modified in Step 10).

Step 12: Now it gives Cipher-Block.

Step 13: Make 4-bits Left-Circular-Shift in key-block i.e. k^{th} 256-bit block.

Step 14: Go to Step 6 and follow the above steps for all remaining blocks except k^{th} block which is key-block.

Step 15: Cipher-Text is ready when all blocks are processed.

3.2 Decryption Process

Phase 1: Decryption

Step 1: Cipher-Text is divided into 'n' 256-bit blocks. Assume counter=n mod 32.

Step 2: The middle block i.e. k^{th} block is the key-block.

Step 3: Make 4-bits Right-Circular-Shift in the key-block i.e. k^{th} 256-bit block.

Step 4: Take n^{th} 256-bit block.

Step 5: Apply bitwise XOR on Step 4 Block and Key-block i.e. k^{th} block.

Step 6: Apply DRDP method on Resultant Block of Step 5 and corresponding (counter) character in key-block.

Step 7: Apply swap operation is as shown in Fig. 2.

Step 8: Apply DRDP character converting technique on Resultant 256-bit block of step 7.

Step 9: Perform 2 Round Operations on *Step 8* Resultant 256-Block in the following way:

- Divide 256-bit Block into 4-bit chunks. Make odd numbered chunks Right Circular Shift.
- Apply DRDP Character Converting technique.

Step 10: Decrement 'n' by 1, calculates counter=n mod 32 and go to Step 3 (Continue the same up to first 256-bit block except k^{th} 256-bit block).

Step 11: Delete all trailing '0's in the final binary-data.

Step 12: Now compressed data is ready.

Phase 2: De-Compression of binary-data into alphabetical words

Step 1: Read the binary bits (word) up to a space.

Step 2: Ignore the binary word if it is a decimal number or a decimal number prefixed by a "0C (form feed)".

Step 3: If length of binary word is multiple of '5' then get the original characters using table1.

Step 4: If length of word is not multiple of '5' then check the value of first 2-bits.

- If it is '0 & 1' then First character is Upper-Case otherwise the entire word is Upper-Case. Now excluding the first 2-bits, convert the remaining bits into characters using the above table.
- If it is '1 & 0' then the entire word is Upper-Case. Now excluding the first 2-bits, convert the remaining bits into characters using the above table.

Step 5: If length of the binary word is 2-bytes or 1-byte, convert it into text.

Phase 3: Dictionary-based Decompression

Step 1: Take the word

Step 2: If length of the word is greater than 3 and word is new, enter this word in temporary Dictionary.

Step 3: If the word is an Integer and not following by a word "0C (form feed)", search the temporary Dictionary and replace the Integer with the corresponding word which is in temporary Dictionary.

Step 4: If the word is an Integer and prefixed by a word "0C (form feed)", do not change the word (Integer).

Step 5: Continue from Step 1 to Step 4 for remaining words. Go to Step 1.

IV. ANALYSIS

In this Section, the proposed algorithm ASCCA is analysed. The proposed system is analysed with the help of the following environment:

The CONFIGURATION of the Computer System where this proposed algorithm has been executed:

- Processor: Intel Core 2 Duo E7500@2.93GHz
- RAM: 2 GB
- Operating System: MS Windows XP
- Hard-Disk: 500 GB
- Java software Jdk 1.5

The following advantages have been identified from the proposed system ASCCA:

1) ASCCA is simple in nature and secure form timing attacks. Here the similar Blocks will take different timings for encryption/decryption. Because the Round-Key is different for different Rounds and Blocks.
2) It provides more security for Data. Because it generates different Round-Keys for the similar types of Blocks in encryption/decryption.
3) In ASCCA, DRDP Character Converting Technique provides strong diffusion by creating the nonlinearity in message.
4) In ASCCA, the Round-Key and Final-Key generation provides strong confusion.
5) Here we can't apply and compare with method "STANDARD FRENQUENCY DISTRIBUTION for ENGLISH"
6) Here the Round-Keys are dynamically generated for each round. So not possible to predict the Round-Keys as well as Final-Keys.
7) Performance analysis between DES, TDES, AES, IDEA & ASCCA (new) as shown in table III.

TABLE III. Performance analysis of DES, TDES, AES, IDEA & ASCCA (new)

Algorithm	**Enc-Time(ms)**
DES	1543
TDES	1701
AES	1387
IDEA	1484
ASCCA(New)	1680

V. SECURITY LEVEL

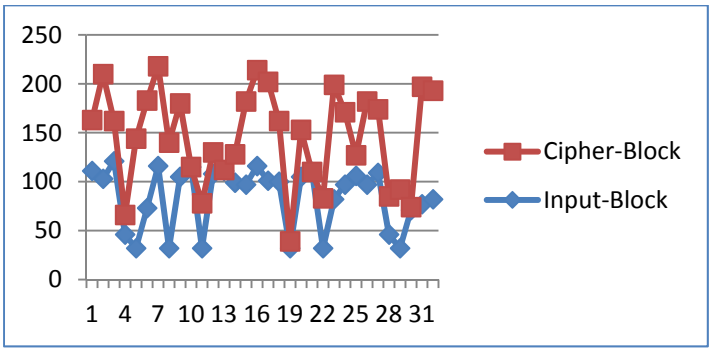
In proposed method, all the characters in the sentence are converted based on Double Reflecting Data Perturbation Method (DRDP). The 256-bit data is processed in 2 Rounds. The converted Data go to Swap and DRDP. Here the privacy of data is measured by the variance between the actual and the perturbed values which is given by the following formula:

$$A = \frac{\mathrm{VAR}(\mathrm{A} - \mathrm{A}')}{\mathrm{VAR}(\mathrm{A})}$$

It has been analysed that the privacy or the security level of the confidential data is improved a lot by the proposed method for Encryption and Decryption [14,15]. The relation between Intermediate-Text and Cipher-Text is as shown in table IV (Some Input-Block):

TABLE IV. Conversion of Temporary Data and Cipher Data

Input-Block	111	103	121	46	32	73	116	32	105	115	32	108	111	99	97	116
	101	100	32	105	110	32	82	97	106	97	109	46	32	71	77	82
Cipher-Block	52	107	41	20	112	110	102	108	75	0	46	22	1	29	85	98
	101	62	7	48	0	51	117	74	21	85	65	39	60	3	120	111



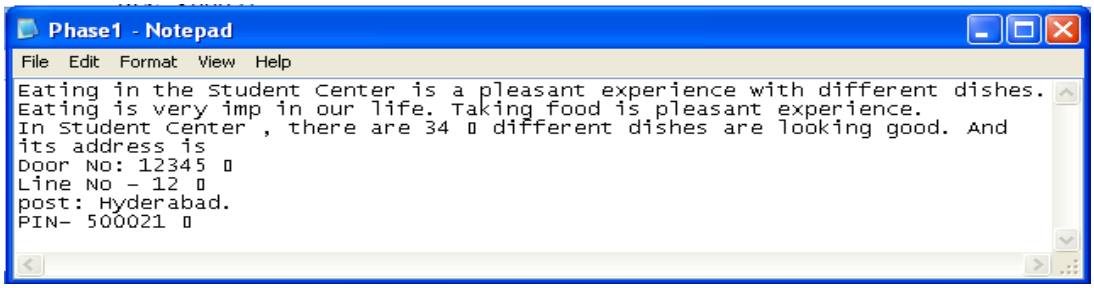
The Graph Representation of Table IV

VI. ILLUSTRATION

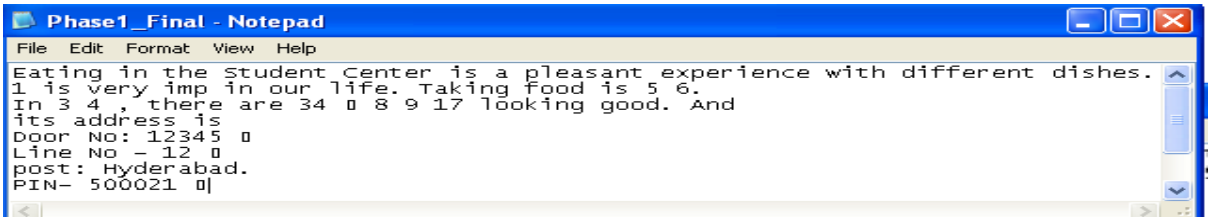
Input Data: "**Eating in the Student Center is a pleasant experience with different dishes.**
Eating is very imp in our life. Taking food is pleasant experience.
In Student Center , there are 34 different dishes are looking good. And
its address is
Door No: 12345
Line No - 12
post: Hyderabad.
PIN- 500021"

Encryption:

Phase 1: The above Input Data is converted like the following:



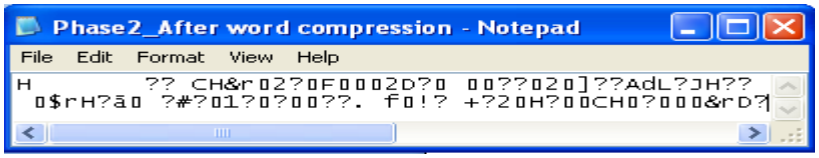
The final conversion is like the following:



And the Dictionary is like the following:

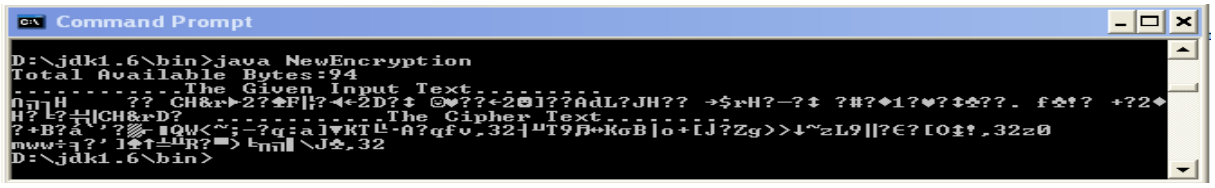
Phase 2:

Here each alphabetical word is compressed and the above Data is converted like the following:



Phase 3:

In this phase, the Cipher-Text will be generated based on figure1 as follows:



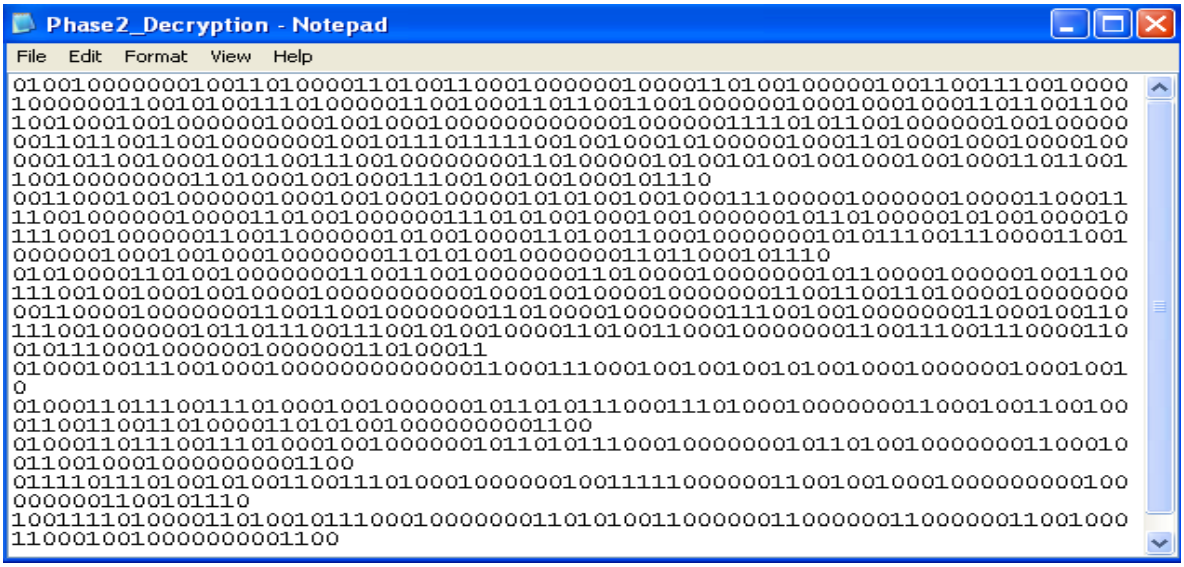
Decryption:

Phase 1:

The Cipher-Text is converted into Intermediate-Text like the following:

Phase 2:

In this phase, the Intermediate-text is divided into sentences based on the binary value of dot (.). Then each sentence is divided into words based on the binary value of space. Now these binary values are converted into text using the technique which was explained in Phase 2 in Decryption. After phase 2, the binary file is like the following:



Phase 3:

In this phase, the temporary dictionary is created. The integers followed by "0C(form feed)" is replaced with the word from dictionary. And the final result is as follows:

"**Eating in the Student Center is a pleasant experience with different dishes.**

Eating is very imp in our life. Taking food is pleasant experience.

In Student Center , there are 34 different dishes are looking good. And

its address is

Door No: 12345

Line No - 12

post: Hyderabad.

PIN- 500021"

VII. CONCLUSION

This Research Paper proposes a new encryption and decryption technique which is based on SMART DICTIONARY BASED ENCODING and DECODING (SDBED). It is suitable to handheld devices. This algorithm provides security to the Data/SMS in handheld devices. It uses both the compression techniques and Cryptography algorithms for providing security to the data in handheld devices. It provides infinite number of Round-Keys for processing Encryption and Decryption process. It uses very fundamental mathematical operations like bitwise XOR and DRDP character conversion technique. It supports Confusion and Diffusion. It also provides more security to the Data. Comparing with other symmetric key cryptography algorithms like DES, TDES, AES, IDEA, the proposed algorithm takes more time for encryption and decryption. But it depends on the number of Rounds in both Encryption and Decryption process. When number of Rounds reduced then Encryption and Decryption Time will be reduced. But it will maintain consistent Security-Level. In near future, this technique will be extended to MMS and Images.

REFERENCES

[1] http://en.wikipedia.org/wiki/Communication.

[2] H. Cheng and X. Li, "Partial encryption of compressed images and videos", IEEE Transactions On Signal Processing, Vol. 48. 8, pp. 2439-2451, August 2000.

[3] E. Celikel and M. E. Dalkilic, "Experiments on a secure compression algorithm", Proceedings of the International Conference on Information Technology: Coding and Computing, vol. 2, pp 150-152, April 2004.

[4] M. Ito, N. Ohnishi, A. Alfalou and A. Mansour, "New image encryption and compression method based on independent component analysis", IEEE information an-d communication technologies from theory to application, pp 1-6, April 2008.

[5] G. H. Keat, A. Samsudin, Z. Zainol, "Enhanced performance of secure image using wavelet compression" World Academy of Science, Engineering and Technology, Universiti Sains Malaysia, pp. 633-636, 2007.

[6] N. V. Thakur, and O. G. Kakde, "Compression mechanism for multimedia system in consideration of information security" Proceeding of International workshop on machine intelligence research, pp 87-96, 2009.

[7] Prakash Kuppuswamy , Dr. Saeed Q Y Al-Khalidi, "Implementation Of Security Through Simple Symmetric Key Algorithm Based On Modulo 37", International Journal of Computers & Technology, ISSN: 2277-3061, Volume 3, OCT 2012.

[8] Ayushi, "A Symmetric Key Cryptographic Algorithm" International Journal of Computer Applications (0975-8887), Volume 1, 2010.

[9] Suyash Verma, Rajnish Choubey, Roopali Soni, "An Efficient Developed New Symmetric Key Cryptography Algorithm For Information Security" , July 2012.

[10] Monika Agarwal, Pradeep Mishra, "A Modified Approach For Symmetric Key Cryptography Based On Blowfish Algorithm", August 2012.

[11] http://protea.worldonline.co.za/fibon.htm.
[12] http://milan.milanovic.org/ math /english/fibo/ fibo3.html.
[13] A. Viji Amutha Mary, Dr. T. Jebarajan, A NovelData Perturbation Technique with higher Security, IJCET, 3(2): 126-132 (2012).
[14] BALAJEE MARAM, Dr CHALLA NARASIMHAM, "Double Reflecting Data Perturbation Methodfor Information Security", OJCST, Dec' 2012, Vol:5, No.2, Pgs: 283-288
[15] http://www.cse.unr.edu/~bebis/CS302/image_info.html.

AUTHOR PROFILE

K.Lakshmanarao is a PhD student at North Orissa University, Baripada, Orissa. He is working as a research scholar under the guidance of Prof. HimaBindu M. He received his MTech in Computer Science and Technology (CSE) from Andhra university, Visakhapatnam, AP, India, in September 2009. His research interests include Computer Networks (especially Wireless Sensor Network) and Information Security. He may be reached at lakshmanarao.k@gmrit.org.

Maringanti HimaBindu received Doctorate (Ph.D.) Artificial Intelligence from Indian Institute of Information Technology, Allahabad, India in 2009. She has worked with BHABHA Atomic Research Institute, ISM, Dhanbad, IIIT, Allahabad. Presently she is working as a Professor of Department of Computer Applications North Orissa University, India. Her research areas of interest include Artificial Intelligence, Image Processing and Pattern Recognition, Natural Language Processing and Cognitive Science, Computer networks and Information security. She has published around 50 papers in national and international conferences and journals. She is the review board member of various reputed journals. She is board of studies member for various autonomous institutions and universities. She can be contacted by email profhbnou2012@gmail.com.

Maram Balajee, working as a Sr.Asst. Prof in Dept. of CSE, GMR Inst. Of Tech., Rajam, AndhraPradesh, INDIA. Obtained his degrees in M.E (CSE) from Anna University, Chennai,Tamilnadu. He also completed MBA(Systems) and MA(MCJ) from Alagappa University, Tamilnadu. Published 15 research papers in International / National Journals / Conferences.