

Ultra Encryption Standard (UES) Version-III: Advanced Symmetric Key Cryptosystem With Bit-level Encryption Algorithm

Satyaki Roy

Department of Computer Science (Autonomous), St. Xavier's College, Kolkata, India
Email: unrivaledsatyaki@gmail.com

Navajit Maitra

Department of Computer Science (Autonomous), St. Xavier's College, Kolkata, India
Email: ruttu04@gmail.com

Shalabh Agarwal

Department of Computer Science (Autonomous), St. Xavier's College, Kolkata, India
Email: shalabh@sxccal.edu

Joyshree Nath

A.K Chaudhuri School of IT, Raja Bazar Science College, Calcutta University, Kolkata, India
Email: joyshreenath@gmail.com

Asoke Nath

Department of Computer Science (Autonomous), St. Xavier's College, Kolkata, India
Email: asokejoy1@gmail.com

Abstract— In the present paper a new cryptographic method called UES Version-III has been introduced. Nath et al recently developed few efficient encryption methods such as UES version-I, Modified UES-I, UES version-II, TTJSA, DJMNA Nath et. al showed that TTJSA and DJMNA is most suitable methods to encrypt password or any small message. The name of the present method is Ultra Encryption Standard Version-III. It is a Symmetric key Cryptosystem which includes multiple encryption, bit-wise randomization, new advanced bit-wise encryption technique with feedback. In this paper, the authors have performed encryption entirely at the bit-level to achieve greater strength of encryption. In the result section the authors have shown the spectral analysis of encrypted text as well as plain text. The spectral analysis shows that UES-III is free from standard cryptography attack such as brute force attack, known plain text attack and differential attack.

Index Items— UES-I, UES-II, Randomization, Bit-wise, Feedback.

I. INTRODUCTION

Due to massive growth in communication technology and the tremendous growth in internet technology in the last decade, it has become a real challenge for a sender to send confidential data from one computer to another. The security of data has now

become a big issue in data communication network. It is not a difficult job for a hacker to intercept that mail and retrieve the question paper if it is not encrypted. Breaking weak password is now not a problem. Public softwares are available to decode password of some unknown e-mail. The data must be protected from any unwanted intruder otherwise a massive disaster may take place. Suppose an intruder intercepts the confidential data of a company and sells it to a rival company, then it will cause great damage to the company from where the data has been intercepted. Cryptography algorithms are of two types (i) Symmetric key cryptography where we use single key for encryption and decryption purpose. And (ii) Public key cryptography where we use one key for encryption purpose and one key for decryption purpose.. Both the methods have their advantages as well as disadvantages. Nath et al. had developed some advanced symmetric key algorithm [1-5],[9-12].

In this paper, the authors have attempted to take encryption one step further by introducing bit-level encryption. The algorithm provides the combined strength of bit-wise randomization and a new module Advanced Bit-wise Encryption Technique with Feedback. We have tested this method on various types of known text files and we have found that, even if there is repetition in the input file, the encrypted file contains no repetition of patterns. Undoubtedly, it provides stronger encryption than the byte-level

encryption method attempted so far. From the detailed test results and analysis shown in the paper, we shall elucidate how this method performs multiple-encryption based on maximum 64-byte password provided by the user.

II. ALGORITHM- UES III

The UES Version- III algorithm includes a number of modules that may be largely classified under two algorithms (i) Bit-wise Randomization and Integration (ii) Advanced Bit-wise Encryption Technique with Feedback (ABETF). The first algorithm uses employs permutation of the plain text bits and the second applies feedback to encrypt the plain bytes beyond recognition.

A. Bit-wise Randomization and Integration

ENCRYPTION:

Step 1: Enter the names of the plain text file, cipher file and the password that may have a maximum length of 64-bytes.

Step 2: Calculate $l = \text{length}(\text{secret key})$.

Step 3: Calculate $\text{cod} = \sum [\text{key}[i] * (i+1)]$, where i is the index value of array 'key' ($0 < i < l$).

Step 4: Calculate $\text{enc} = \text{mod}(\text{cod}, 60)$ and $\text{rand} = \text{mod}(\text{cod}, 20)$ where $\text{enc} = \text{encryption number}$ and $\text{rand} = \text{randomization number}$. If $\text{enc} < 10$ then $\text{enc} = 10$. If $\text{rand} < 10$ then $\text{rand} = 10$.

Step 5: Invoke bit-wise encryption on the plain file with feedback with the ABETF algorithm.

Step 6: Compute $l = \text{sizeof}(\text{plain text file})$.

Step 7: Compute $\text{count} = (\text{siz} * \text{siz}) / 8$ where $\text{siz} = 64$. It signifies the number of bytes encrypted at once. By default $\text{count} = 512$ bytes.

Step 8: Compute $z = l / \text{count}$. It signifies the number of iterations that are needed to encrypt a file.

Step 9: Create key matrix 'mat' of dimension $\text{siz} \times \text{siz}$. It holds the values 1 to $[(\text{siz} * \text{siz}) - 1]$ row-wise.

Step 10: Define $s = 0$

Step 11: If $s \geq \text{enc}$ then GOTO step-26

Step 12: Randomize the key matrix 'mat' by MSA algorithm as many as 'rand' times.

Step 13: Extract count bytes of plain file and split them into respective bits and store in 1d array 'parr'.

Step 14: Define auxiliary array 'parr2'. Define $i = 0$.

Step 15: If $i \geq (\text{siz} * \text{siz})$ then GOTO step-18

Step 16: Define $\text{ro} = i / \text{siz}$, $\text{co} = \text{mod}(i, \text{siz})$ and $n = \text{mat}[\text{ro}][\text{co}]$ where n is the position corresponding to position i of the plain bits array.

Step 17: Perform $\text{parr2}[i] = \text{parr}[n]$ in order to randomize the bits of the plain file by exchanging the

bits according to the randomized key matrix. Increment i . GOTO step-15

Step 18: Convert the bits in the array parr2 into corresponding bytes. GOTO step-13

Step 19: Now the algorithm processes the residual bytes. The residual bytes are again split into bits and stored in 1d array 'parr'.

Step 20: Randomize the residual key matrix 'matt' using the module rant which takes the number of residual bytes 'rem' as parameter. We perform *leftshift, cycling, downshift, chaining operations* on the key matrix 'matt'. The chaining operation is shown below.

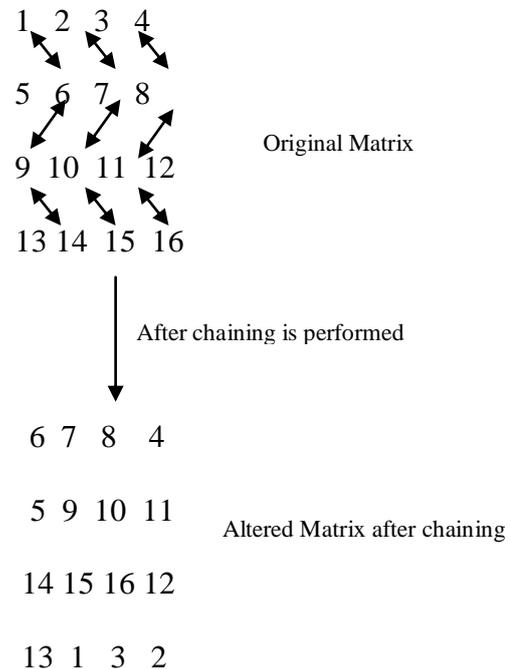


Fig 1: Chaining operation

Step 21: The size of the array 1d array 'parr' = $(\text{rem} * 8)$ where $\text{rem} = \text{number of residual bytes}$. Define $i = 0$.

Step 22: If $i \geq (\text{rem} * 8)$ then GOTO step-24. Define $\text{ro} = i / 8$, $\text{co} = i \% 8$ and $n = \text{matt}[\text{ro}][\text{co}]$.

Step 23: Perform $\text{parr2}[i] = \text{parr}[n]$ in order to randomize the bits of the plain file where parr2 is again the auxiliary array of plain bits. Increment i . GOTO step-22.

Step 24: Convert the bits into the respective bytes to yield residual cipher bytes. Increment s . GOTO step-11.

Step 25: END

B. Advanced Bit-wise Encryption Technique with key pad and Feedback (ABETF algorithm)

ENCRYPTION:

Step-1: Compute $\text{cod} = \sum \text{key}[i] * (i+1)$ from the password 'key' provided by the user.

- Step-2: Compute $k = \text{modulus}(\text{cod}, 256)$. Define $i=0$.
- Step-3: Write the character with ASCII value k in the file containing the feedback keypad. Increment k and i . If $i < 1$ GOTO 3.
- Step-4: Randomize the feedback keypad using simple character randomization.
- Step-5: Split the plain and feedback keypad into respective bits and store them in two files.
- Step-6: Extract one bit from the plain file and the feedback file each and store them in characters 'ch' and 'chb' respectively. Define $c=0$.
- Step-7: Compute $m = (ch + chb + c) - 96$
- Step-8: If $m \geq 2$ then perform $m = m - 2$.

Table-I: ABETF ENCRYPTON (FEEDBACK GENERATION): The algorithm takes into consideration that the ASCII value of '0' is 48. Hence, the subtraction of $(2 * 48)$ is performed during the computation of m . After the bitwise OR Cipher bit becomes the feedback and the carry is ignored

| | | | | |
|---------------|---|---|---|---|
| FEEDBACK: c | 0 | 1 | 1 | 0 |
| CIPHER: ch | 1 | 1 | 0 | 0 |
| KEY BITS: chb | 1 | 0 | 0 | 1 |
| PLAIN BITS: | 0 | 0 | 1 | 1 |

- Step-9: Perform $c = m$
- Step-10: Write the integer m in the output file.
- Step-11: Goto 6 until the end of the file is reached.
- Step-12: Convert the bits in the output file into respective bits to produce the cipher file.
- Step-13: END

DECRYPTION:

- Step-1: Compute $\text{cod} = \sum \text{key}[i] * (i+1)$ from the password 'key' provided by the user.
- Step-2: Compute $k = \text{modulus}(\text{cod}, 256)$. Define $i=0$.
- Step-3: Write the character with ASCII value k in the file containing the feedback keypad. Increment k and i . If $i < 1$ GOTO 3.
- Step-4: Randomize the feedback keypad with bit-wise randomization.
- Step-5: Split the plain and feedback keypad into respective bits and store them in two files.
- Step-6: Extract one bit from the plain file and the feedback file each and store them in characters 'ch' and 'chb' respectively. Define $c=0$.
- Step-7: Compute character $chb = (ch + chb + c) - 96$
- Step-8: If $chb \geq 2$ then perform $chb = chb - 2$.
- Step-9: Perform $c = ch - 48$

Table-II: ABETF DECRYPTION (the cipher bit is feedback)

| | | | | |
|----------------|---|---|---|---|
| FEEDBACK: c | 0 | 1 | 1 | 0 |
| PLAIN BITS: ch | 0 | 0 | 1 | 1 |
| KEY BITS: chb | 1 | 0 | 0 | 1 |
| CIPHER BITS: | 1 | 1 | 0 | 0 |

- Step-10: Write the ASCII of character 'chb' in the output file.
- Step-11: Goto 6 until the end of the file is reached.
- Step-12: Convert the bits in the output file into respective bits to produce the cipher file.
- Step-13: END

III. DIAGRAMMATIC REPRESENTATION OF THE WORKING OF UES-III.

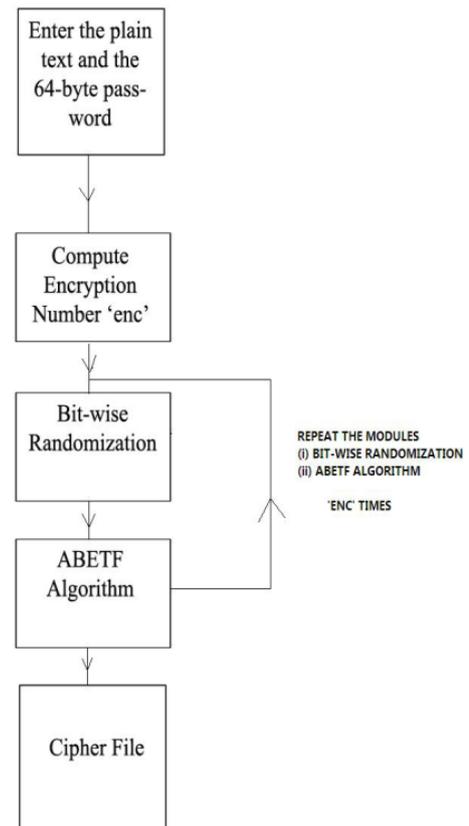


Figure II: The working of UES III

IV. TEST RESULTS

In the present paper, the authors have combined the two modules of bit-level randomization and Advanced Bit-wise Encryption Technique with feedback. The test results have been recorded with great care to ensure that the algorithm not only works for every file format but also yields satisfactory test results for all possible file sizes.

The frequency analysis in result-II corresponds to the plain text of 512 characters of 'A'. Generally, for such inputs of same characters, certain patterns of characters are recorded in the cipher file. However, as it may be evident from the graph of Figure-III, no such patterns have been noticed.

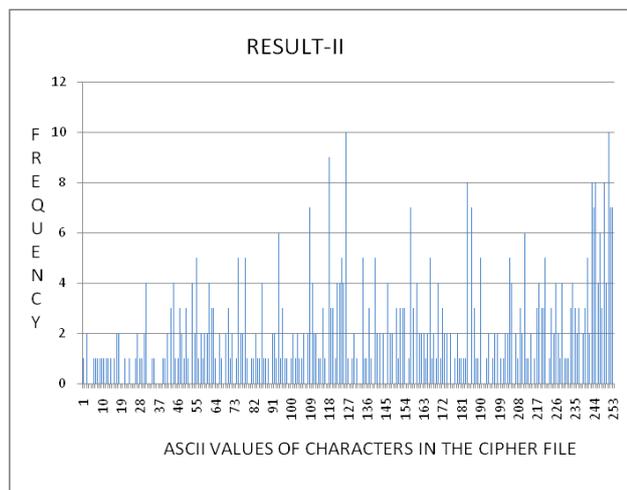


Figure-IV: RESULT-II (shown below) - The graph representing the frequency of each character (of ASCII 0-255) in the cipher file. The plain text has 512 characters of 'A'.

V. DISCUSSIONS ON FUTURE SCOPE AND CONCLUSION

In the previous endeavours of UES Version-I, UES Modified Version-I and UES Version-II, the authors have worked exclusively on bytes. However, in the third module, the entire encryption process has been performed at the bit-level. The plain text files have been split into respective bits before applying the aforementioned algorithms. From the test results shown before, it is evident that the algorithm takes care of plain text inputs such as ASCII 0 or 1. Even when the same characters are provided as input, the cipher files have almost no occurrence of repetitive patterns. The bit feedback module has been employed for the very first time, to incorporate bit-level encryption with automatic feedback generation. The use of multiple encryption and the role of the password provided by the user have also been demonstrated in the test result IV (ii).

The results show that this method is too hard to break by using any kind of brute force method. As mentioned before have applied our method on some known text where the single character repeats itself for a number of times and we have found that after encryption there is no repetition of pattern in the output file. Moreover, it must be remembered, if the cipher file is tampered and certain character(s) in the file get altered, it would be impossible to retrieve the plain file, since the feedback generated will be different for different characters.

ACKNOWLEDGEMENT

We are grateful to the Department of Computer Science for giving us the unique opportunity to work on Symmetric Key Cryptography. One of the authors (AN) sincerely expresses his gratitude to Fr. Dr. Felix Raj and Fr. Jimmy Keepuram for allowing us to carry out research work. AN is thankful to the University Grant Commission for their support and financial assistance. JN is grateful to A.K. Chaudhuri School of IT and SR, NM, SA and AN are thankful to St. Xavier's College. The authors extend their thanks to the Computer Science Hons. (2011-2012) for their encouragement and help.

REFERENCES

- [1] Symmetric Key Cryptography using Random Key generator: Asoke Nath, Saima Ghosh, Meheboob Alam Mallik: Proceedings of International conference on security and management(SAMf10) held at Las Vegas, USA Jull 12-15, 2010), P-Vol-2, 239-244 (2010).
- [2] A new Symmetric key Cryptography Algorithm using extended MSA method :DJSA symmetric key algorithm, Dripto Chatterjee, Joyshree Nath, Suvadeep Dasgupta and Asoke Nath : Proceedings of IEEE CSNT-2011 held at SMVDU(Jammu) 3-5 June,2011, Page-89-94.
- [3] New Symmetric key Cryptographic algorithm using combined bit manipulation and MSA encryption algorithm: NJSSAA symmetric key algorithm: Neeraj Khanna,Joel James,Joyshree Nath, Sayantan Chakraborty, Amlan Chakrabarti and Asoke Nath : Proceedings of IEEE CSNT-2011 held at SMVDU(Jammu) 03-06 June 2011, Page 125-130.
- [4] Advanced Symmetric key Cryptography using extended MSA method: DJSSA symmetric key algorithm: Dripto Chatterjee, Joyshree Nath, Soumitra Mondal, Suvadeep Dasgupta and Asoke Nath, Journal of Computing, Vol3, issue-2, Page 66-71, Feb(2011).
- [5] Advanced Steganography Algorithm using encrypted secret message : Joyshree Nath and

Asoke Nath, International Journal of Advanced Computer Science and Applications, Vol-2, No-3, Page-19-24, March(2011).

- [6] Symmetric key Cryptography using modified DJSSA symmetric key algorithm, Dripto Chatterjee, Joyshree Nath, Sankar Das, Shalabh Agarwal and Asoke Nath, Proceedings of International conference Worldcomp 2011 held at Las Vegas, USA, July 18-21, Page 312-318, Vol-I(2011).
- [7] Cryptography and Network, Willian Stallings, Prentice Hall of India.
- [8] Cryptography & Network Security, B.A.Forouzan, Tata Mcgraw Hill Book Company.
- [9] An Integrated symmetric key cryptography algorithm using generalized vernam cipher method and DJSA method: DJMNA symmetric key algorithm, Debanjan Das, Joyshree Nath, Megholova Mukherjee, Neha Chaudhury and Asoke Nath, Proceedings of IEEE conference WICT-2011 held at Mumbai University Dec 11-14,2011
- [10]Symmetric key cryptosystem using combined cryptographic algorithms-generalized modified vernam cipher method, MSA method and NJJSAA method: TTJSA algorithm, Trisha Chatterjee, Tamodeep Das, Joyshree Nath, Shyan Dey and asoke Nath, Proceedings of IEEE conference WICT-2011 held at Mumbai University Dec 11-14,2011.
- [11]Ultra Encryption Standard(UES) Version-I: Symmetric Key Cryptosystem using generalized modified Vernam Cipher method, Permutation method and Columnar Transposition method, Satyaki Roy, Navajit Maitra, Joyshree Nath,Shalabh Agarwal and Asoke Nath, Proceedings of IEEE sponsored National Conference on Recent Advances in

Communication, Control and Computing Technology-RACCCT 2012, 29-30 March held at Surat, Page 81-88(2012).

- [12]Symmetric key Cryptography using two-way updated – Generalized Vernam Cipher method: TTSJA algorithm, International Journal of Computer Applications(IJCA, USA), Vol 42, No.1, March, Pg: 34 -39(2012).

Satyaki Roy has recently completed graduation in Computer Science Honours from St. Xavier's College(Autonomous), Kolkata. He is currently working in cryptography at bit-level and have already published UES Version-I which he undertook as B.Sc. Project work.

Navajit Maitra has recently completed graduation in Computer Science Honours from St. Xavier's College(Autonomous), Kolkata. His B.Sc Project was UES version-I which already published in a National conference. Currently he working in cryptography.

Joyshree Nath has completed her M.Tech(IT) from Calcutta University. She has been actively involved in research work in the field of cryptography, Steganography.

Shalabh Agarwal is the Assistant Professor at St. Xavier's College(Autonomous), Kolkata. His field of research includes green computing , e-learning, cryptography and steganography. He has published many papers in National and International Journals and conferences.

Asoke Nath is the Assistant Professor at St. Xavier's College(Autonomous), Kolkata. He is involved in research work in the area of symmetric key cryptography, asymmetric key cryptography, Steganography, Green Computing, e-learning methodologies, Distance education methodologies. He has published many research papers from International Conferences and Journals. He has given invited tutorial on Introduction to Cryptography and Network security in National and International conferences.