RASS- A Concurrency Based Bitwise Symmetric Key Cryptographic Algorithm

Abhriya Roy, Ronit Ray, Saptarshi De, Shalini Guha, Sukalyan Goswami, Ratan Kumar Basak, Bipasha Mukhopadhyay, Souvik Chatterjee, Amrin Zaman, Sucheta Nag

Department of Computer Science and Engineering,
University of Engineering and Management, Kolkata, India

royabh314@gmail.com | ray.ronit@gmail.com | saptarshide2013@gmail.com | shaliniguha2@gmail.com | sukalyan.goswami@gmail.com | ratan.basak@uem.edu.in | bipasha.mukherjee@uem.edu.in | souvik.chatterjee@uem.edu.in | amrin.zaman03@gmail.com | suche1996@gmail.com

Contents

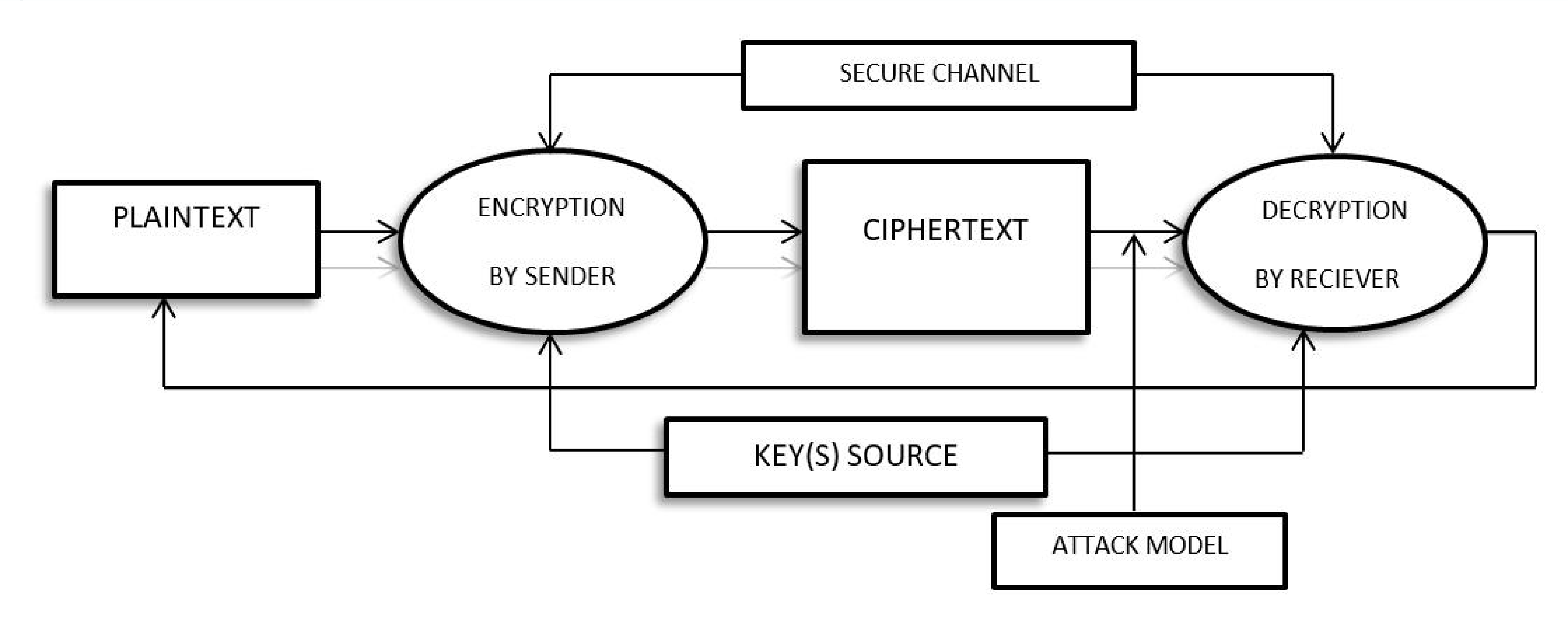
Abstract

- **Features of the proposed algorithm :**
  - Bit-level security
  - Linear time complexity
  - Robust security (using two 16 bit keys)
  - Concurrency
  - Multithreading

Introduction

- Cryptography is the science of secrets; an art of changing plain, readable text messages to encrypted ciphertext which can only be read by the intended receiver.
- Key terms :
  - Plaintext
  - Ciphertext
  - Algorithms involved:
    - Encryption algorithm
    - Decryption algorithm
  - Secret Key

Introduction (Contd.)



Objectives of Modern Cryptography

- Confidentiality:
  - Only an authorized person can access the information being communicated
  - A unique key is maintained between the sender and the receiver
- Integrity:
  - The information cannot be manipulated or tampered with
  - Helps in constructing a secure channel for data communication
- Non-repudiation:
  - One cannot deny his/her involvement in the creation of data or its transmission
- Authentication:
  - The sender and receiver can be assured of the other's identity and the origin/destination of the information

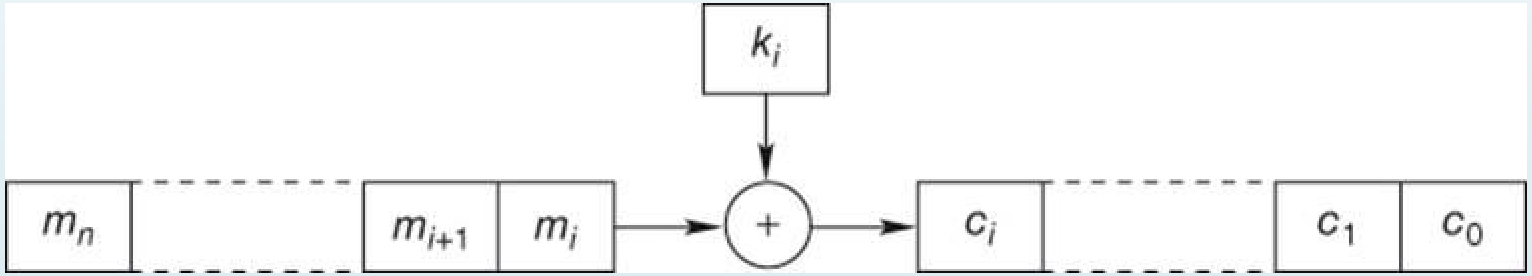
Background

- Bitwise Ciphers
- XOR function
- One-Time Pad and Key Security
- Types of Modern Cryptography
- Common Symmetric Cryptography Algorithms
- Concurrency, Parallelism, Multithreading

Background

- Bitwise Cipher:

A bitwise cipher manipulates (performs an operation on) each bit of data sequentially instead of entire blocks.



Background (Contd.)

The XOR function/operation:

The XOR operator returns a 1/HIGH/TRUE whenever the inputs do not match, which occurs when one of the two inputs is exclusively true. This is the same as addition modulo 2.

Bitwise XOR (⊕)		
OPERAND 1(OP1)	OPERAND 2(OP2)	RESULT= OP1 ⊕ OP2
0	0	0
0	1	1
1	0	1
1	1	0

The XOR function/operation (Contd.)

- A unique property of XOR is that $(A \oplus B) \oplus A$ returns B, because:

$A \oplus B \oplus A$

$\Rightarrow A \oplus A \oplus B$ (by commutative law)

$\Rightarrow (A \oplus A) \oplus B$

$\Rightarrow 0 \oplus B$

$\Rightarrow B$

This is vital to the process of cryptography.

While this won't give you the individual plaintexts, because XOR is not directly reversible, it still gives you a considerable amount of information about each of them.

Key Security And The One-Time Pad

- One-Time Pad:

A randomly generated key of same length as plaintext is XORed with the plaintext, bit by bit.

- Issues with One-Time Pad:
  - Same key could be generated twice
  - XORing of the two ciphertexts with the same could return considerable information about the two plaintexts for interceptor

$$
\begin{array}{ll}
C1 \oplus C2 & \\
(P1 \oplus K) \oplus (P2 \oplus K) & \text{(k is the same)} \\
P1 \oplus K \oplus P2 \oplus K & \text{(removing parentheses)} \\
P1 \oplus P2 \oplus (K \oplus K) & \text{(commutative law)} \\
P1 \oplus P2 \oplus 0 & \\
P1 \oplus P2 &
\end{array}
$$

  - Crib dragging: Guessing of the plaintext by passing small sequences along the ciphertext
  - Overhead increases due to the transmission of the huge key, so small keys are used instead

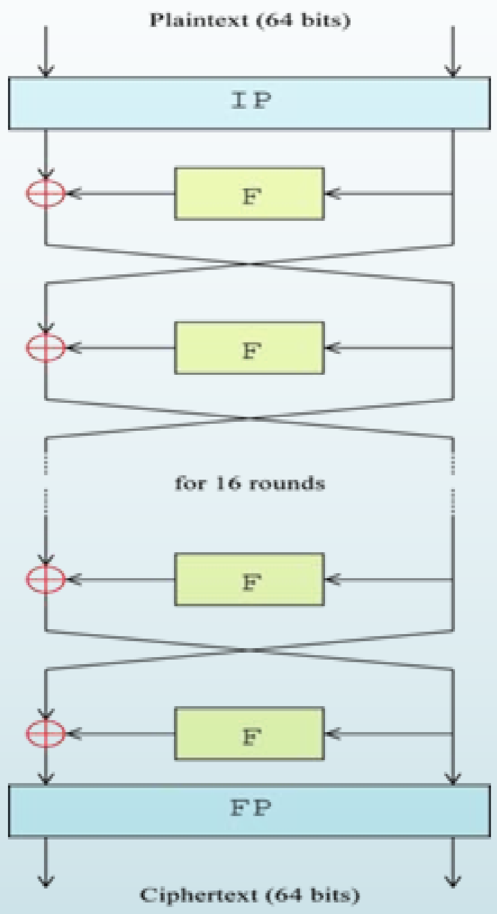
Types of Modern Cryptography

- Secret Key Cryptography or Symmetric Cryptography
  - Encryption: Plaintext to ciphertext using a secret key
  - Decryption: Ciphertext to plaintext using verified key
  - Key Generator: A pseudo-random function generates the key to share

- Public Key Cryptography or Asymmetric Cryptography
  - Asymmetric: Key information held by the sender and the receiver are dissimilar or asymmetric
  - Different Keys: One party has the secret key, other has the public key for decrypting it

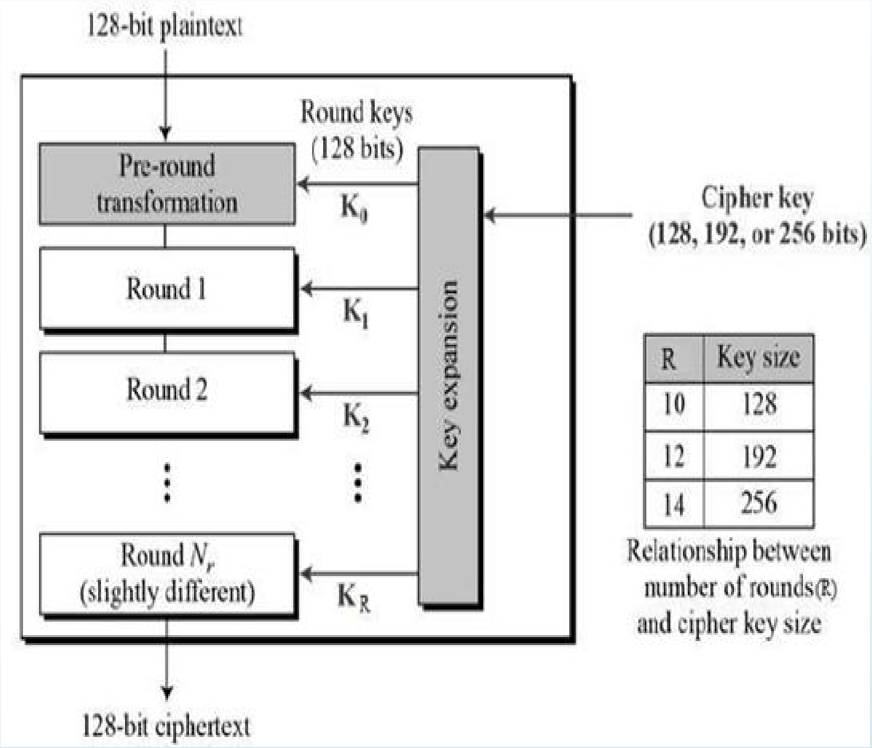
Some Symmetric Cryptographic Algorithms

- DES(Data Encryption Standard)
  - DES (Data Encryption Standard) is a symmetric key 64-bit block cipher encryption algorithm developed by IBM.
  - Has an effective key size of 56 bits and another 8 bits is used for checking parity
  - Performs 16 rounds to encrypt data.
  - Disadvantage:
    - The small key size compared to other encryption algorithms made it vulnerable to various attacks, therefore making it insecure.



Some Symmetric Cryptographic Algorithms (Contd.)

- AES (Advanced Encryption Standard):
  - Symmetric key 128 bit block cipher encryption algorithm.
  - Has variable key lengths of 128,192 and 256 bits.
  - Performs 10,12 and 14 rounds to encrypt 128,192 and 256 bits of data respectively.
  - Most effective and widely used encryption scheme.



Concurrency, Parallelism and Multithreading

- Concurrency
  - Makes progress on more than one task at the same time
  - No concept of finishing one task before the next
- Parallelism
  - Splitting of one task into several small sub-tasks which are then run parallely
- Multithreading
  - Allows multiple flows of control to a processor
  - Implements concurrency
  - Each of these flows is called a thread
  - Increases the efficiency of the algorithm

Previous Work

- Comparative research (see References)
- K. Naveen Kumar et al
  - Bit-Level
  - Feistel-like approach
  - Bitwise shifts, logical XOR, addition-subtraction modulo 32
- Satyaki Roy et al
  - Bit-Level
  - Randomized bits, feedback mechanism
- Ashoke Nath et al
  - Bit-Level
  - Total inversion, bit manipulation at random prime locations, with bitwise shifts
- Rajdeep Chakraborty et al
  - Bit-level
- Desoky and Madhusudhan
  - Plaintext scrambling by division into 8 different planes and applying a separate key on each of the planes implementing a modified version of the Hill Cipher

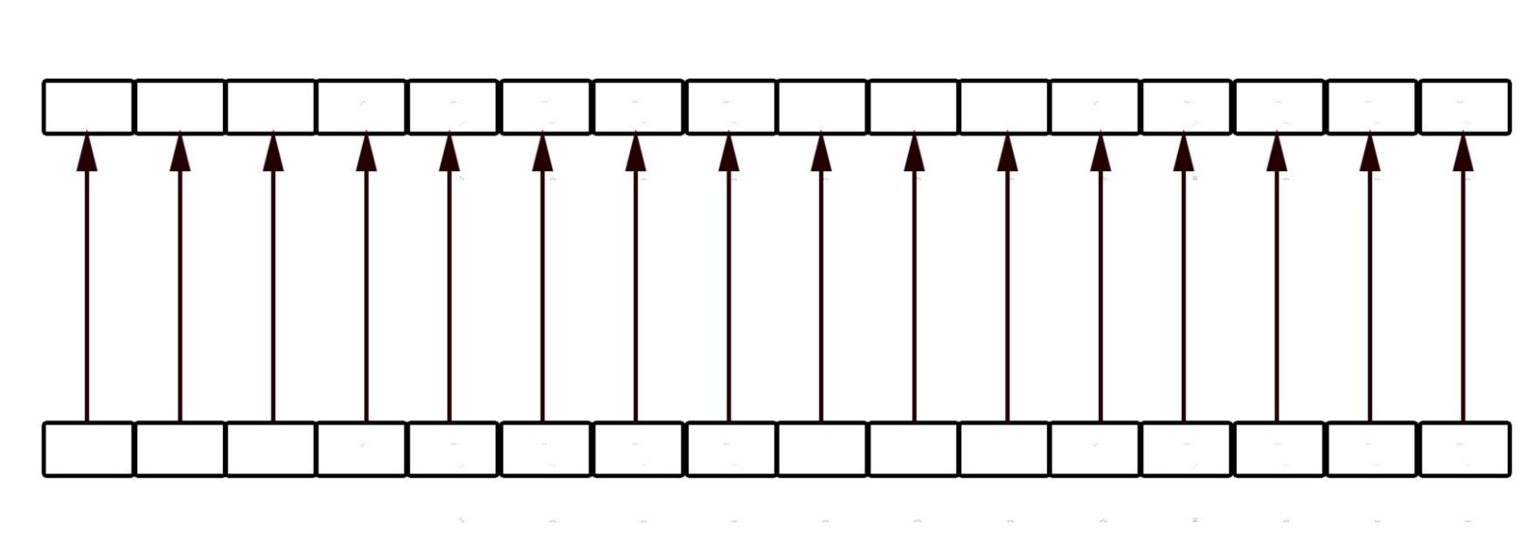
Proposed Algorithm

1. Partitioning of plaintext into two parts.
2. A separate thread operates on each part concurrently.
3. Creation of two 16-bit keys.
   1. Each key applied to a separate part of the plaintext
   2. 2^32 possible permutations over 2^16 for single key.
4. Scrambling by applying different schemes on different 16-bit blocks. Increases complexity.

   Operation:
   1. Linear (for even).
   2. A crisscross (for odd).
      1. First half of plaintext with second half of key and vice versa.

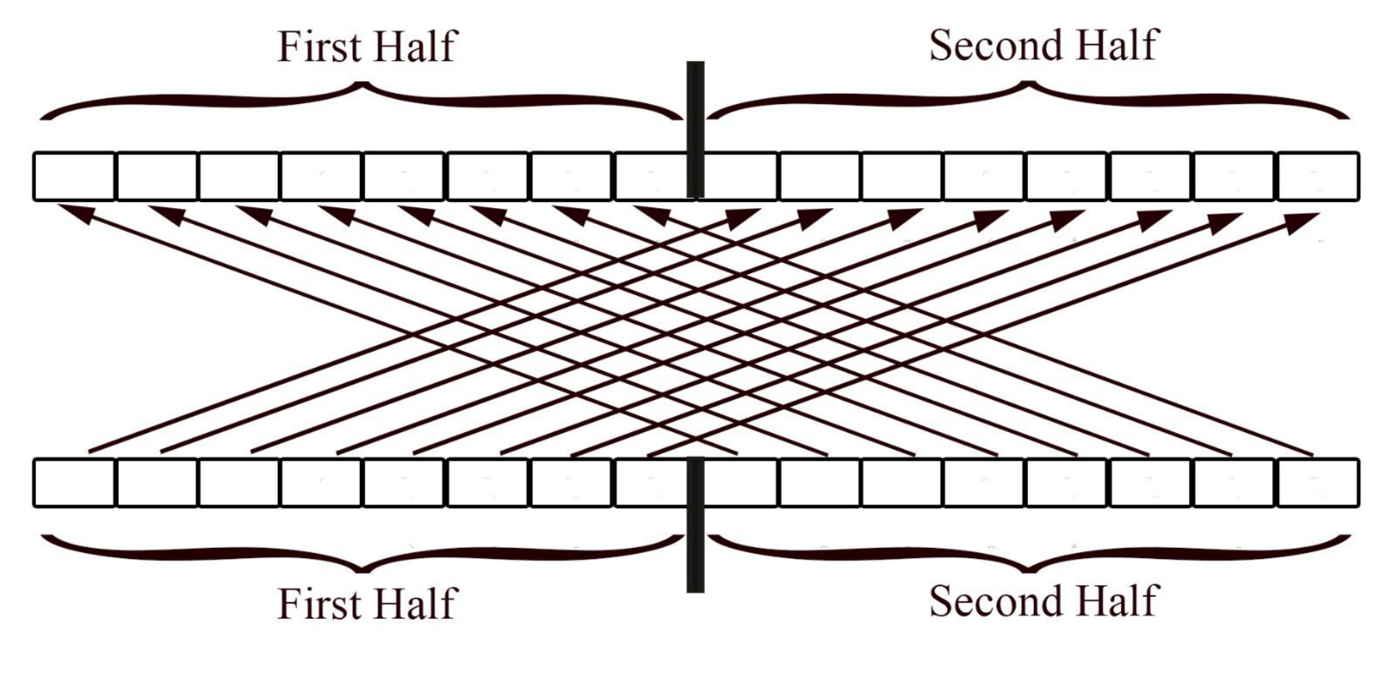
Linear XORing

- Same pattern as One-Time Pad.
- Each plaintext bit XORed with the corresponding key bit.



Criss-Cross XORing

- Divide the 16-bit block into two 8-bit halves.
- XOR the first half of plaintext with second half of key.
- XOR the second half of plaintext with first half of key.



Algorithm: Key Generation

```
PROCEDURE GenerateKey(length)
START
STEP 1. SET key ← ""
STEP 2. for all i between 0 and length -1
STEP 3.       num ← random floating-point no. between 0 & 1
STEP 4.       IF (num > 0.5)
              THEN key ← key+ '1'
          ELSE
           key ← key + '0'
STEP 5. Return key
STOP.
```

Algorithm: XOR Methods

PROCEDURE EVEN(Text, Key)
//both params are 16 bits in length
START
STEP 1. SET encrypted ← “”
STEP 2. for all i in Text
STEP 3. encrypted+= Text[i] ⊕ Key[i]
STOP.
	PROCEDURE ODD(Text,Key)
	//both params are 16 bits in length
	START
	STEP 1. SET encrypted ← “”
	STEP 1. for all i in Text
	//crisscross done below
	STEP 2. if(0<i<8)
	STEP 3. encrypted+= Text[i] ⊕ Key[i+8]
	else if(8<=i<16)
	STEP 4. encrypted+= Text[i] ⊕ Key[i-8]
	STOP.

Algorithm: Encryption

INPUT: Plaintext, Key1, Key2

OUTPUT: Ciphertext (same length as plaintext)

BEFORE EXECUTION: Build plaintext blocks of 16-bit width each.

```
START
STEP 1. SET half ← length_of_plaintext/2
STEP 2. for all i in BLOCKS
STEP 3.    If (0<i<half)
STEP 4.       If (i is EVEN) THEN
STEP 5.          ciphertext+=EVEN(BLOCKS[i] , Key1)
                else
STEP 6.          ciphertext+=ODD(BLOCKS[i] , Key1)
             else
STEP 7.       If (i is EVEN) THEN
STEP 8.          ciphertext+=EVEN(BLOCKS[i] , Key2)
                else
STEP 9.          ciphertext+=ODD(BLOCKS[i] , Key2)
STOP.
```

Algorithm: Decryption

INPUT: Ciphertext, Key1, Key2.

OUTPUT: Plaintext (same length as ciphertext)

BEFORE EXECUTION: Build ciphertext blocks of 16-bit width each.

```
START
STEP 1. for all i in BLOCKS
STEP 2.     If (0<i<half)
STEP 3.         If (i is EVEN) THEN
STEP 4.             ciphertext+=EVEN(BLOCKS[i] , Key1)
                   else
STEP 5.             ciphertext+=ODD(BLOCKS[i] , Key1)
               else
STEP 6.         If (i is EVEN) THEN
STEP 7.             ciphertext+=EVEN(BLOCKS[i] , Key2)
                   else
STEP 8.             ciphertext+=ODD(BLOCKS[i] , Key2)
STOP.
```

Performance Analysis and Comparison

- The proposed algorithm is a bitwise cipher or “bit cipher”
  - it is unlike any of the block cipher standards like AES, DES or Blowfish, and is hence incomparable to any of them.
- It can, however, be compared with the algorithm proposed by K. Naveen Kumar et al, which is also faster that Satyaki Roy et al.
- Asoke Nath et al and Rajdeep Chakraborty et al failed to provide execution times for their proposed bit-level cryptographic algorithms and hence were not admissible to the comparison

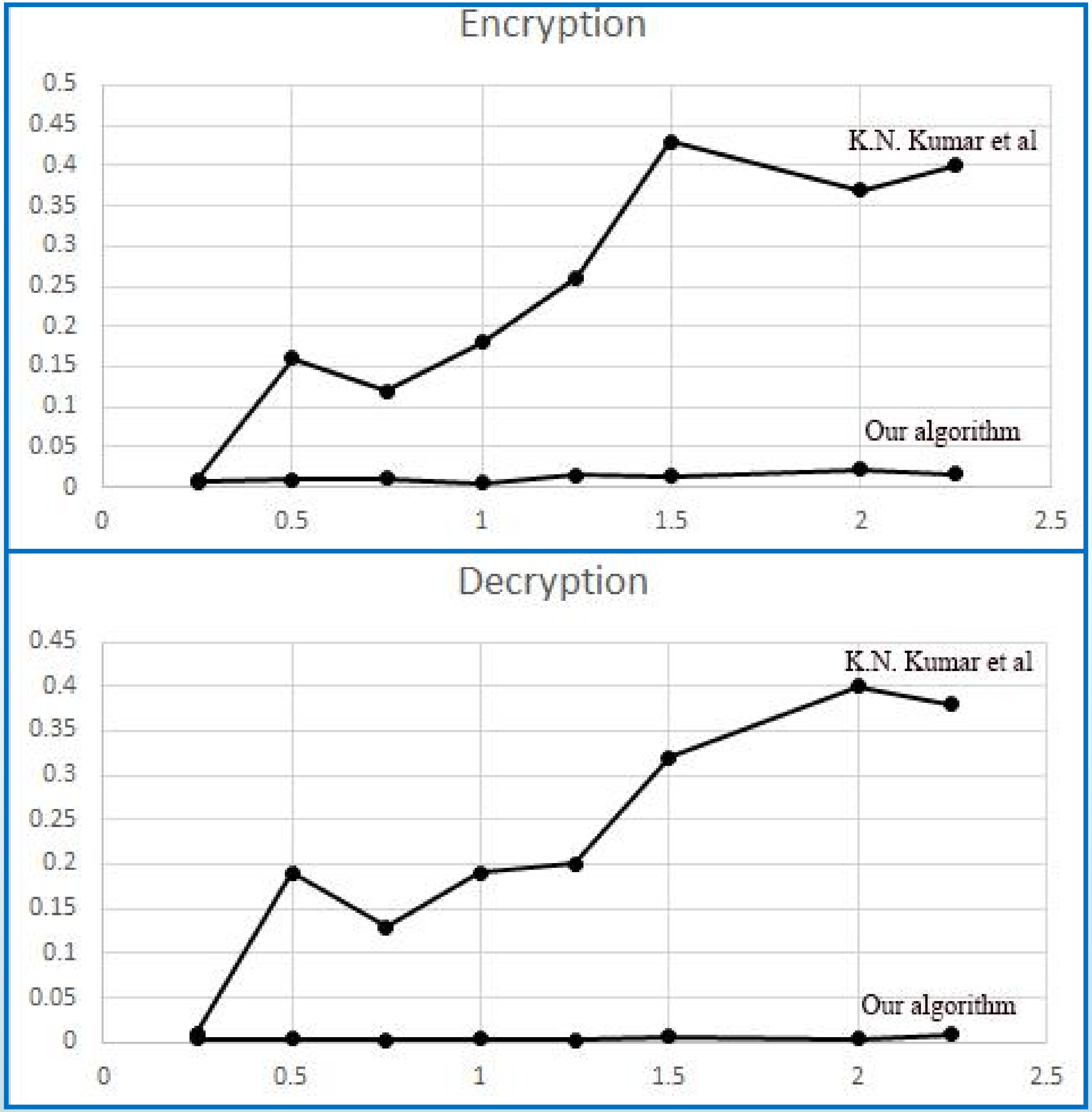
Proposed v/s KN Kumar et al

Note: KN Kumar et al only provided data for files up to 2.25 KB in size.

File Size (KB)	Encryption Time (s)		Decryption Time (s)	
	Our Algorithm	K.N. Kumar et al	Our Algorithm	K.N. Kumar et al
0.25	0.0062	0.01	0.004	0.1
0.5	0.0091	0.16	0.003	0.19
0.75	0.0113	0.12	0.0027	0.13
1	0.0052	0.18	0.0035	0.19
1.25	0.0147	0.26	0.0028	0.2
1.5	0.0143	0.43	0.0061	0.32
2	0.0226	0.37	0.004	0.4
2.25	0.016	0.4	0.0086	0.38

Trends

Time taken for growing data size



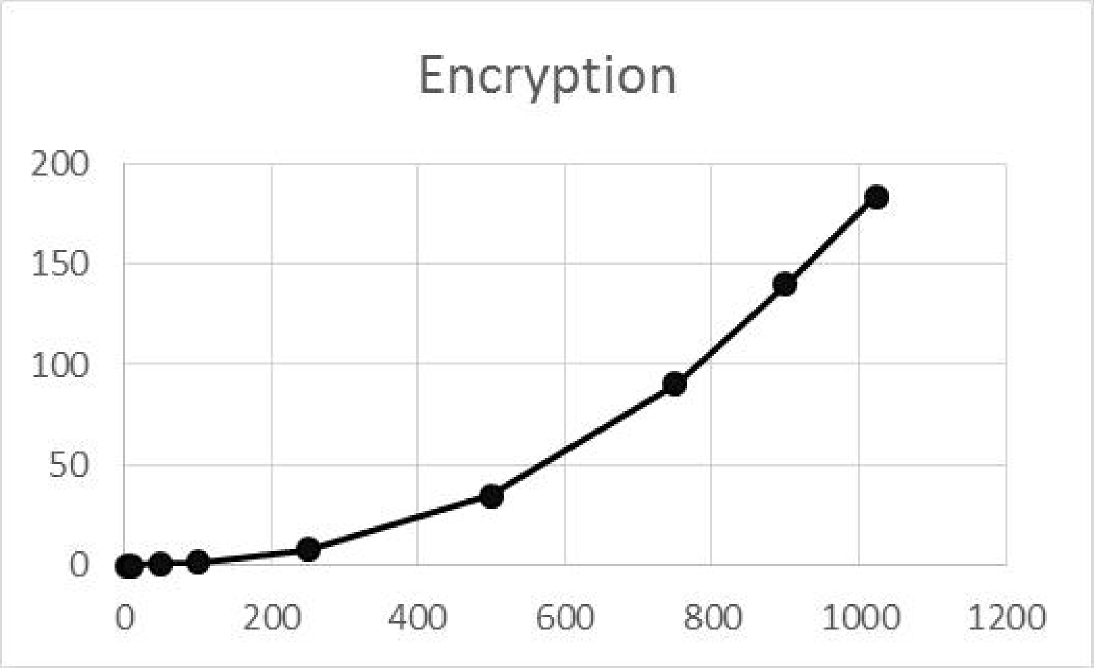
Further Trends

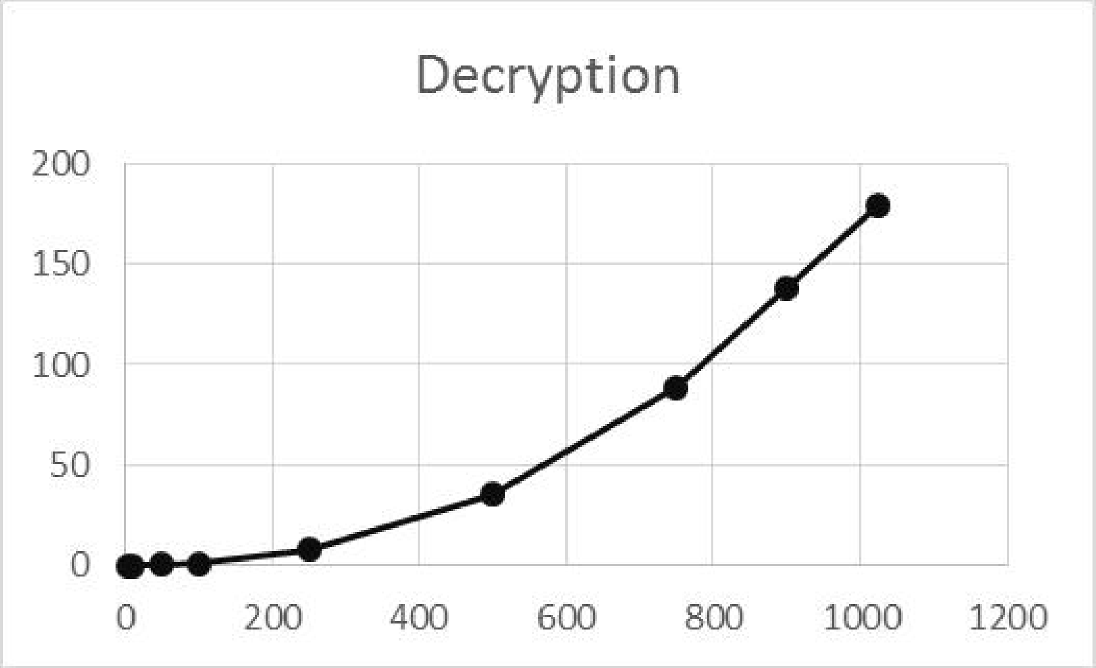
For data sizes up to 1MB.

File Size (KB)	Encryption Time (s)	Decryption Time (s)
5	0.102	0.019
10	0.164	0.044
50	0.869	0.401
100	1.238	1.235
250	7.807	7.73
500	34.682	35.431
750	90.087	88.839
900	139.87	138.14
1024	183.516	178.923

Growth Rate

All background CPU processes aside, the general nature of the graph's growth rate is linear, or O(n), which is in keeping with the algorithm defined previously.





Conclusion

- In this paper, we have proposed a symmetric key cryptographic algorithm with the following features:
- Two 16-bit keys for reinforced security, working on two separate parts of the plaintext.
- Two different encryption schemes for alternate blocks of text, thus increased unpredictability and improved security.
- Concurrently working on two parts of plaintext at the same time, thus optimizing running time.
- Providing security at the bit-level for assured security throughout the data life cycle.

The proposed algorithm performs significantly better than any similar proposed schemes in the past and has a linear growth rate.

Frequently Asked Questions

1. **What language was the algorithm implemented in, and on what platform?**

A. The algorithm was implemented in Java 8 on Eclipse Mars. Code is available upon request.

All tests were performed on a system with an Intel i7-3520M 2.93 GHz processor and 8 GB RAM.

2. **Why is the proposed algorithm incomparable with AES, DES and Blowfish?**

A. The three algorithms mentioned are all block ciphers, meaning they provide security at block-level. The proposed algorithm is a bitwise or bit-level cipher, which manipulates each individual bit of data and is hence secure down to every bit. This obviously is a different paradigm and is much slower, and hence should not be compared to block ciphers.

References

[1] Vivek Kaushik, Vikram Singh, Manish Vats, “Data Encryption Techniques for Dependable and Secure Cloud Computing”, 2014 IJIRT, Volume 1 Issue 6.

[2] Laurens Van Houtven, Crypto 101, II. Building Blocks, Exclusive OR

[3] Mihir Bellare, Phillip Rogaway, Introduction to Modern Cryptography, Chapter 4, Symmetric Encryption

[4] Piotr Nienaltowski, “Practical Framework for contract-based concurrent object-oriented programming”

[5] K. Naveen Kumar, G.V.S. Raj Kumar, K.T. Praveen Kumar, P. Chandra Sekhar, "Bitwise Operations Based Encryption and Decryption", International Journal on Computer Science and Engineering (IJCSE), Vol 3 No. 1 Jan 2011

[6] S. Vidhya, K. Chitra, "Format Preserving Encryption using Feistel Cipher", International Conference on Research Trends in Computer Technologies(ICRTCT-2013), Proceedings published in International Journal of Computer Applications® (IJCA) (0975–8887)

[7] Murray Eisenberg, "Hill Ciphers and Modular Linear Algebra", November 3, 1999.

[8] Ahmed Desoky, Anju Panicker Madhusoodhan, “Bitwise Hill Crypto System”, 2011 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), pp. 080-085

References (contd.)

[9] Sourabh Chandra, Siddhartha Bhattacharya, Smita Paira, Sk Safikul Alam, "A Study and Analysis on Symmetric Cryptography", 2014 International Conference on Science, Engineering and Management Research.

[10] Gurpreet Singh, Supriya, "A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security", International Journal of Computer Applications (0975-8887) Volume 67- No. 19, April 2013

[11] Sourabh Chandra, Smita Paira, Sk Safikul Alam, Goutam Sanyal, "A Comparative Survey of Symmetric and Asymmetric Key Cryptography", 2014 International Conference on Electronics, Communication and Computational Engineering (ICECCE)

[12] Harsh Mathur, Zahid Alam, "Analysis in Symmetric and Asymmetric Cryptology Algorithm", International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 4, Issue 1, January-February 2015

[13] Satyaki Roy, Shalabh Agarwal, Asoke Nath, Navajit Maitra, Joyshree Nath, "Ultra Encryption Algorithm (UEA): Bit Level Symmetric key Cryptosystem with Randomized Bits and Feedback Mechanism, International Journal of Computer Applications (0975-8887) Volume 49- No.5, July 2012

[14] Asoke Nath, Madhumita Santra, Supriya Maji, Kanij Fatema Aleya, "Bit Level Symmetric Key Encryption Algorithm (BLSKEA-1) Version-1", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 3 Issue 11, November 2015

[15] Rajdeep Chakraborty, Sonam Agarwal, Sridipta Misra, Vineet Khemka, Sunit Kr Agarwal, JK Mandal, "Triple SV: A Bit Level Symmetric Block Cipher Having High Avalanche Effect", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 2, No.7, 2011