Analysis and optimization of Galois/Counter Mode(GCM) using MPI

—

By Pulak Sahoo

Under V S Ananthanarayana

What is this project about?

Introduction

This Project

- AES
- Parallel Computing
- Message Passing Interface
- Galois Counter Mode

Advanced Encryption Standard (AES)

1. Proposed by NIST
2. Encryption algorithms
3. Symmetric block cipher
4. Processes each input block separately
5. Uses same key for both encryption and decryption.
6. Key length can be 128, 192 or 256 bits
7. Block size can only be 128 bits

Parallel Computing

1. Efficiently utilize the hardware resources
2. Two models:
    a. Shared memory
    b. Message passing
3. Single Instruction Multiple Data (SIMD)

Message Passing Interface

1. Writing parallel programs
2. communicates among different processors
3. Communication and synchronization between different processors
4. Requires subroutine calls

Message Passing Interface

1. MPI has three categories of subroutines,
    a. Communication
    b. Synchronization
    c. Enquiries.
2. Communication can be point to point or collective.
3. Barriers can be applied for synchronizations.
4. Enquiries give us information about the number of processes and tags associated with each process.

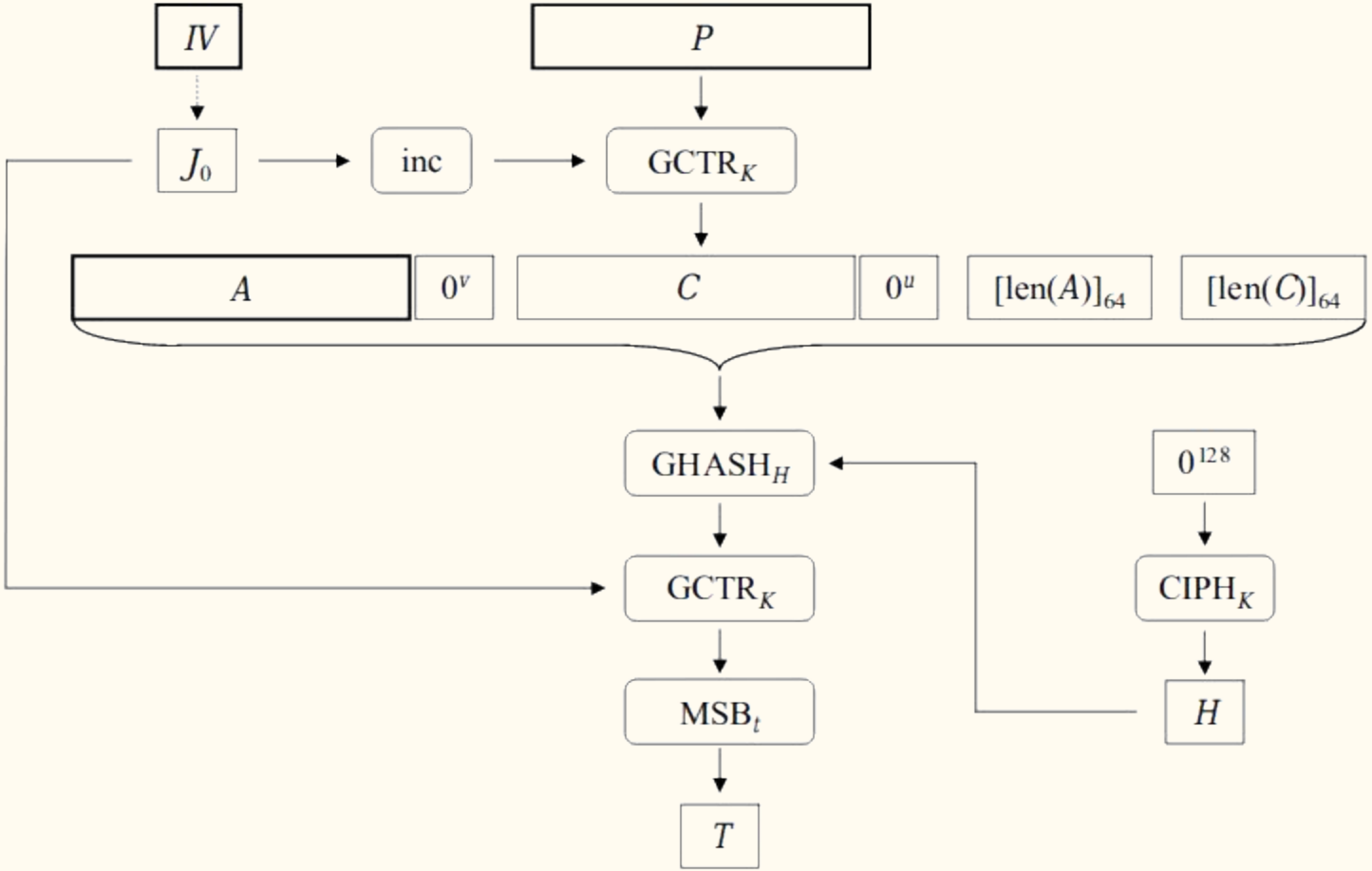
Galois Counter Mode

1. NIST standard SP 800-38D
2. Parallelized to provide:
    a. High speed message authentication
    b. Confidentiality of data.
3. Counter mode of encryption (CTR)

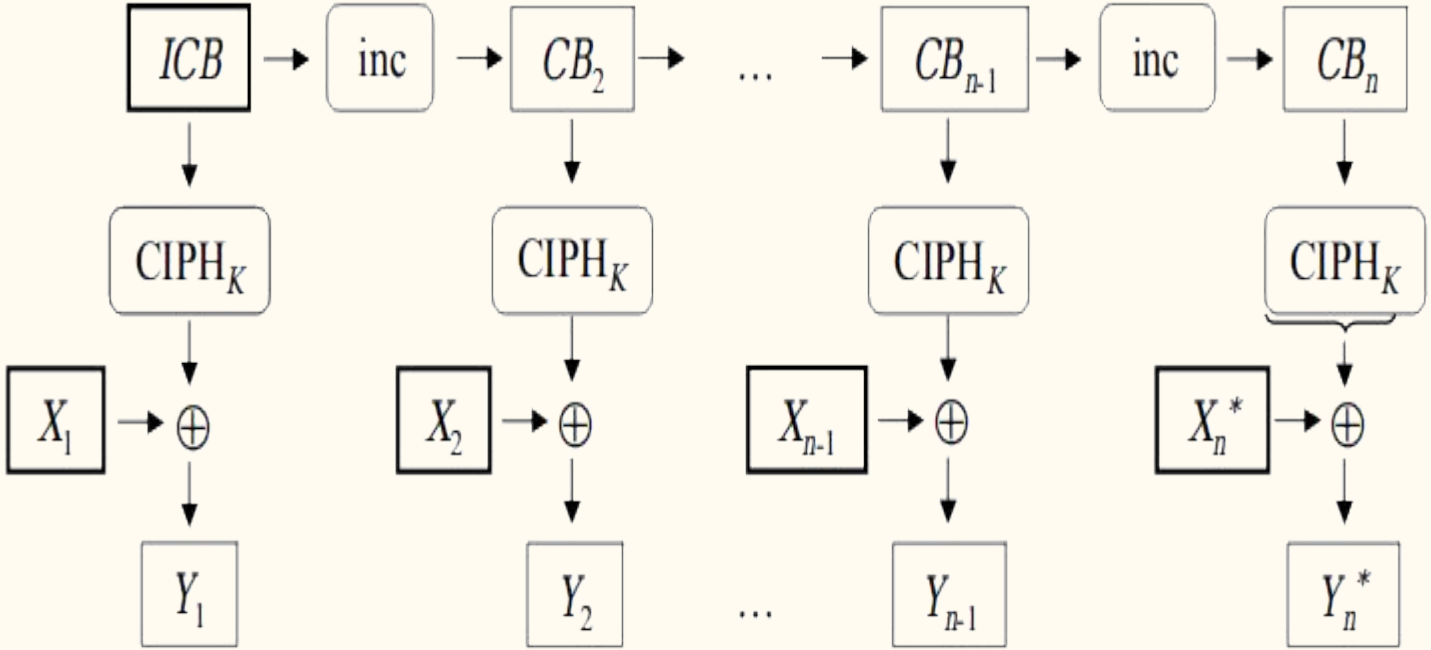
Galois Counter Mode

1. Resulting cipher text is multiplied:
   a. Key material
   b. Message length over binary Galois Field (GF 2128).
2. The CTR mode is multiplied with the universal hashes over binary Galois field (GF 2) also known as Galois Field multiplication.
3. Hashes used in GCM, provides the authenticity of confidential data over GF 2.

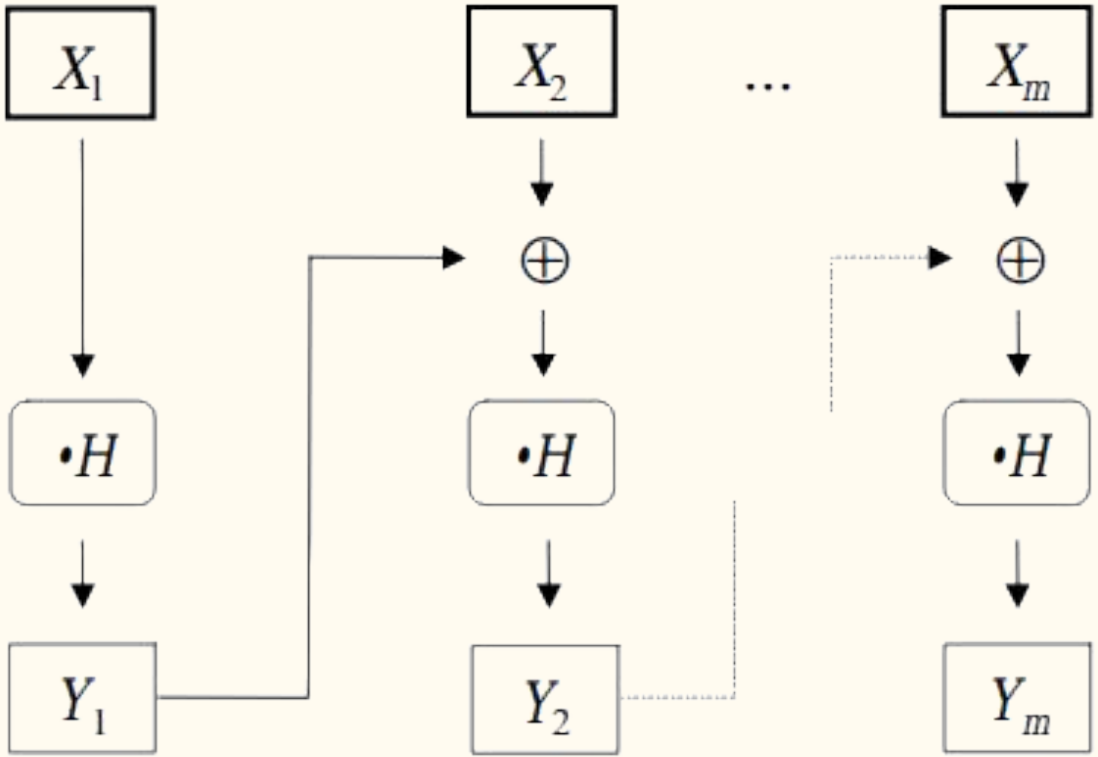
Sequential Algorithm


IV
P
J_0
inc
GCTR_K
A
0^v
C
0^u
$[\mathrm{len}(A)]_{64}$
$[\mathrm{len}(C)]_{64}$
GHASH_H
0^{128}
GCTR_K
CIPH_K
MSB_t
H
T

Parallel GCTR


ICB
inc
CB2
...
CBn-1
inc
CBn
CIPHK
CIPHK
CIPHK
CIPHK
X1
X2
Xn-1
Xn*
Y1
Y2
...
Yn-1
Yn*

Parallel GHash


X_1
X_2
...
X_m
$\oplus$
$\oplus$
$\cdot H$
$\cdot H$
$\cdot H$
Y_1
Y_2
Y_m

Problems

No of times data distributed=$2n_1$

No of times data collected=$2n_2$

Total data exchange= $2n_1 + 2n_2$

Revised Parallel Version

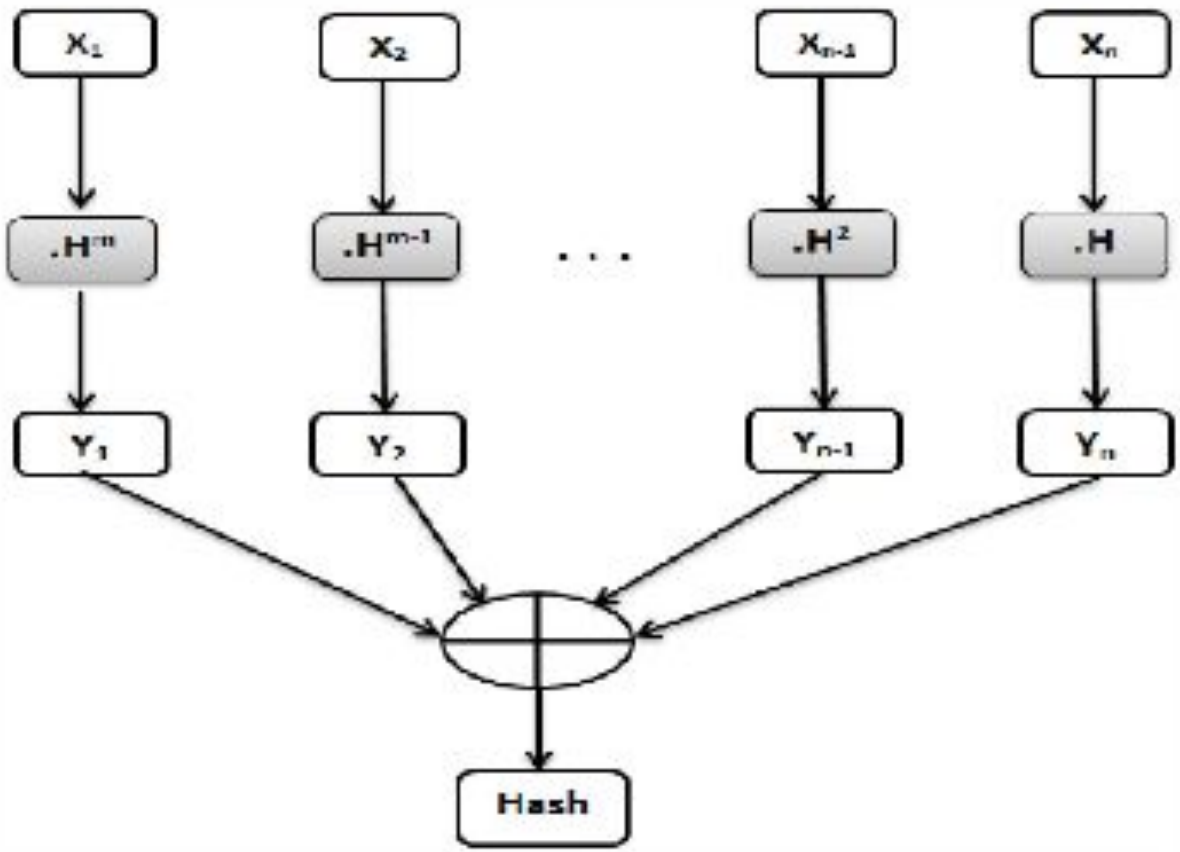
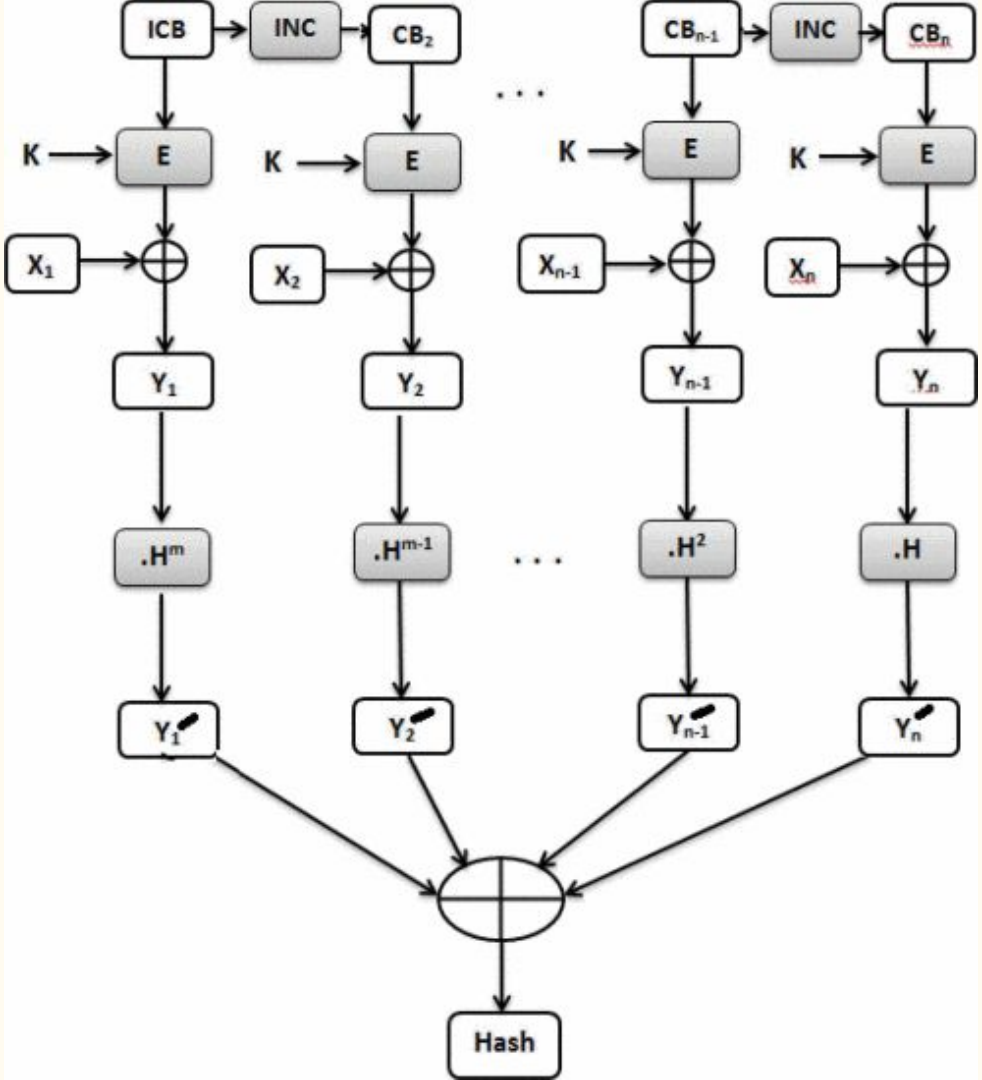


Figure 4. Parallel GHASH


ICB
INC
CB2
CBn-1
INC
CBn
K
E
X1
X2
Xn-1
Xn
Y1
Y2
Yn-1
Yn
.Hm
.Hm-1
.H2
.H
Y1'
Y2'
Yn-1'
Yn'
Hash

Benefits

No of times data distributed= n_1

No of times data collected= n_2

Total data exchange= $n_1 + n_2$

Reduction Factor= 2

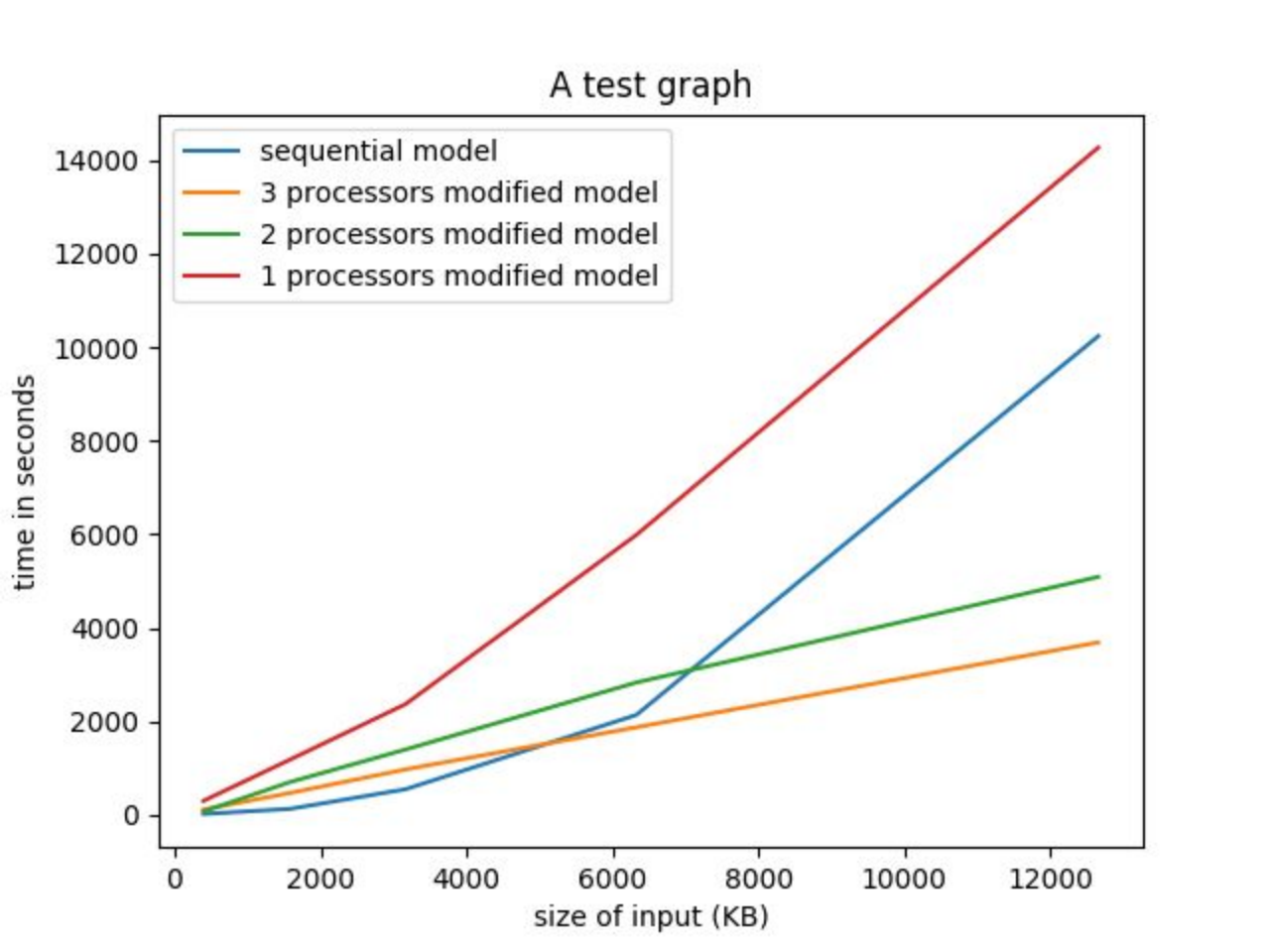
Evaluation

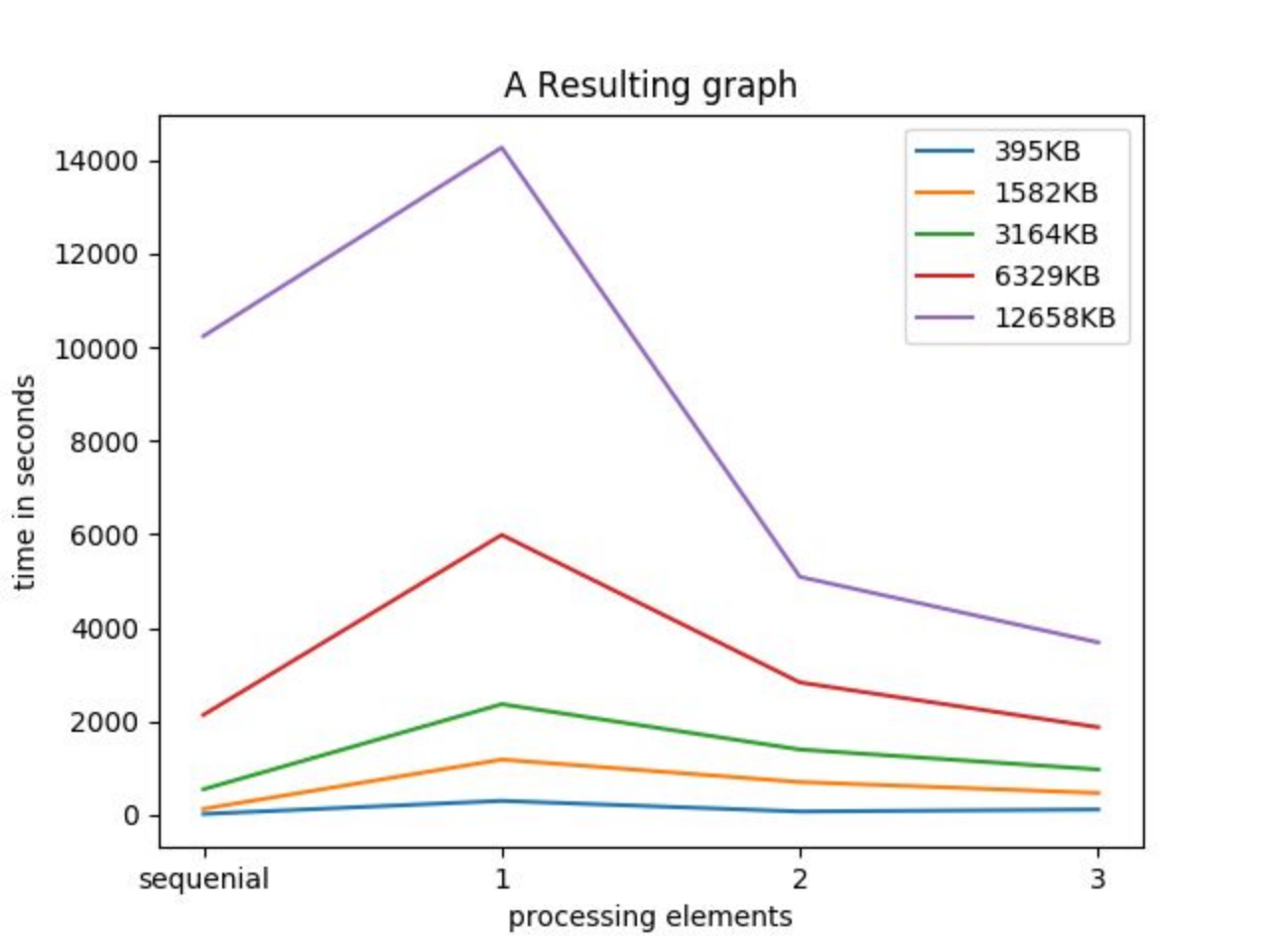
Results

- Time v/s Size of Input
- Time v/s Number of Input

Observed Result

	395 KB	**1582 KB**	**3164 KB**	**6329 KB**	**12658 KB**
Sequential	23 sec	131 sec	551 sec	2140 sec	10234 sec
1 Processor modified	302 sec	1185 sec	2369 sec	5987 sec	14264 sec
2 Processor modified	75 sec	704 sec	1400 sec	2832 sec	5089 sec
3 Processor modified	118 sec	471 sec	974 sec	1876 sec	3689 sec


A test graph
sequential model
3 processors modified model
2 processors modified model
1 processors modified model
time in seconds
size of input (KB)
0
2000
4000
6000
8000
10000
12000
14000


A Resulting graph
14000
12000
10000
8000
6000
4000
2000
0
time in seconds
sequenial
1
2
3
processing elements
395KB
1582KB
3164KB
6329KB
12658KB

Conclusion, Improvement & Future Work

Conclusion

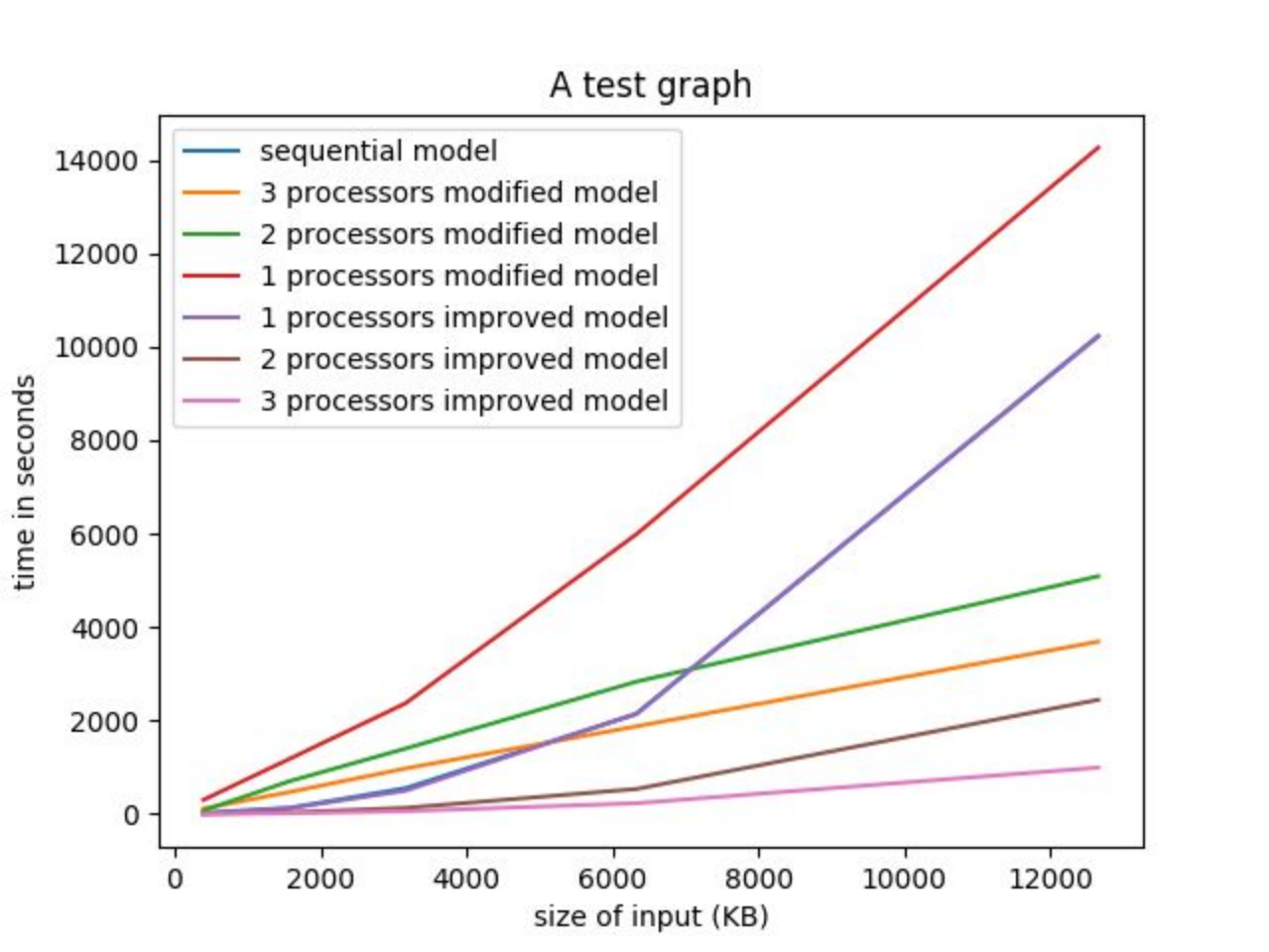
In this paper I implemented and analyzed the GCM algorithm using MPI. Although it seems to be difficult to parallelize GCM using MPI I tried to implement it using a modified version as given in the paper referred as well as an improved version.

Improvement

- Divide into clusters and add it to GCM traditional method.
- Use 256 bit SHA for better authenticity.

Observed Result

	395 KB	1582 KB	3164 KB	6329 KB	12658 KB
Sequential	23 sec	131 sec	551 sec	2140 sec	10234 sec
1 Processor modified	302 sec	1185 sec	2369 sec	5987 sec	14264 sec
2 Processor modified	118 sec	704 sec	1400 sec	2832 sec	5089 sec
3 Processor modified	75 sec	471 sec	974 sec	1876 sec	3689 sec
1 Processor improved	7 sec	123 sec	499 sec	2147 sec	10221 sec
2 Processor improved	2 sec	32 sec	130 sec	534 sec	2444 sec
3 Processor improved	1 sec	15 sec	60 sec	230 sec	994 sec


A test graph
sequential model
3 processors modified model
2 processors modified model
1 processors modified model
1 processors improved model
2 processors improved model
3 processors improved model
time in seconds
size of input (KB)
0
2000
4000
6000
8000
10000
12000
14000

Future

- Counter mode is vulnerable to attacks. Instead of using the counter mode, pseudo random number generator can be used in GCTR to improve security.
- Galois field multiplication can be parallelized using MPI.
- Performance analyses of GCM on clouds.

References

1. M. H. Durad, M. N. Khan and Z. Ahmad, "Analysis and optimization of Galois/Counter Mode (GCM) using MPI," *2015 12th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, Islamabad, 2015, pp. 333-337.
2. William, Stallings, and William Stallings. Cryptography and Network Security, 4/E. Pearson Education India, 2006.

Thank You!