MONASH University
Information Technology

ASSESSMENT COVER SHEET

Student ID number	28498704
Given Name	Siyuan
Family name	Yan

Unit Name and Code:	FIT3143 Parallel Computing
Campus:	Clayton
Assignment Title:	Assignment 2
Name of Lecturer:	Christopher Watkins
Name of Tutor:	Pierce O'Hara-Wild
Tutorial Day and Time:	12:00 Monday
Phone Number:	0450881725
Email Address:	syan0009@student.monash.edu

Has any part of this assignment been previously submitted as part of another unit/course? ☐ Yes ☐ No

Due Date:	Date Submitted:

All work must be submitted by the due date. If an extension of work is granted this must be specified with the signature of the lecturer/tutor.

Extension granted until (date)_____ Signature of lecturer/tutor _____

Please note that it is your responsibility to retain copies of your assessments.

Intentional plagiarism or collusion amounts to cheating under Part 7 of the Monash University (Council) Regulations

Plagiarism: Plagiarism means taking and using another person's ideas or manner of expressing them and passing them off as one's own. For example, by failing to give appropriate acknowledgement. The material used can be from any source (staff, students or the internet, published and unpublished works).

Collusion: Collusion means unauthorised collaboration with another person on assessable written, oral or practical work and includes paying another person to complete all or part of the work.

Where there are reasonable grounds for believing that intentional plagiarism or collusion has occurred, this will be reported to the Associate Dean (Education) or delegate, who may disallow the work concerned by prohibiting assessment or refer the matter to the Faculty Discipline Panel for a hearing.

Student Statement:

- I have read the university's Student Academic Integrity Policy and Procedures.
- I understand the consequences of engaging in plagiarism and collusion as described in Part 7 of the Monash University (Council) Regulations http://adm.monash.edu/legal/legislation/statutes
- have taken proper care to safeguard this work and made all reasonable efforts to ensure it could not be copied.
- No part of this assignment has been previously submitted as part of another unit/course.
- I acknowledge and agree that the assessor of this assignment may for the purposes of assessment, reproduce the assignment and:
  - i. provide to another member of faculty and any external marker; and/or
  - ii. submit it to a text matching software; and/or
  - iii. submit it to a text matching software which may then retain a copy of the assignment on its database for the purpose of future plagiarism checking.
- I certify that I have not plagiarised the work of others or participated in unauthorised collaboration when preparing this assignment.

Signature .....

* delete (iii) if not applicable

The information on this form is collected for the primary purpose of assessing your assignment and ensuring the academic integrity requirements of the University are met. *Other purposes of collection include recording your plagiarism and collusion declaration, attending to course and administrative matters and statistical analyses. If you choose not to complete all the questions on this form it may not be possible for Monash University to assess your assignment. You have a right to access personal information that Monash University holds about you, subject to any exceptions in relevant legislation. If you wish to seek access to your personal information or inquire about the handling of your personal information, please contact the University Privacy Officer: privacyofficer@adm.monash.edu.au*

Design and Implementation of Inter-Process Communication Architecture

Siyuan Yan
Monash University
E-mail: syan0009@student.monash.edu

***Abstract*—Wireless Sensor Network (WSN) is an Inter-Process Communications System. The network can be used to monitor conditions and send data through the network. The report presents a method for simulating the WSN system using Message Passing Interface (MPI) library in C language. To simulate the inter-process communication (IPC) architecture, MPI Cartesian Topology is chosen as it can create new communicators that order the process ranks in a way that can be a better match for the IPC architecture. Besides, to make sure the security of communication, a proposed parallel architecture of Advanced Encryption Standard (AES) algorithm is implemented using C language and OpenMP standard and tested in a single multicore computer. The experiment result shows a speedup greater than 1 in terms of the comparison between the AES algorithm using OpenMP and serial implementation will indicate the success of the parallel implementation.**

Keywords- WSN, MPI, Parallel architecture, IPC, encryption, AES, OpenMP, Master slave, Nearest neighbor communication.

I. INTRODUCTION

Wireless Sensor Network (WSN) is a network for monitoring and recording the physical conditions of the environment and organizing data at a central location [1]. In computer science, WSN is a famous and active research topic with numerous workshops and conferences arranged each year.

The assignment is to simulate a WSN system. Each node in the system communicate with its neighbors and the node will report the event and some details to base station when an event happened in this node. The base station collects all events and records them into a log file. The objective of the assignment is to find an efficient IPC scheme that minimizes messages to the base station whilst satisfying the WSN's event detection criterion. In addition, all events and messages sent should be encrypted.

Inter-process communication (IPC) is a mechanism that allows processes to communicate with each other and synchronize their actions [2]. In this report, an inter-process communication scheme is used to simulate the WSN architecture, an MPI Cartesian topology scheme is used to achieve inter-process communication grid architecture for nodes. The encryption between communication is used called AES algorithm. To make sure the performance of the program, a proposed parallel AES algorithm using OpenMP is used, the original source code is from the link [5].

There are three hypotheses presented

1. The messages per event reported to the base station is less than 4.
2. The scheme chosen can be implemented efficiently and satisfies WSN's event detection criterion.
3. A comparison between AES using OpenMP and serial implementation are evaluated, the speedup is greater than 1.

II. ILLUSTRATION OF IPC GRID ARCHITECTURE

A. Description of Inter-process communication

To simulate the wireless sensor network, an inter-process communication scheme is employed. In this assignment, the WSN system has 20 nodes and a base station. The 20 nodes are arranged in a 4×5 rectangular-shaped grid, shown in Figure 1.

	A	B	C	D	E
1	A1	B1	C1	D1	E1
2	A2	B2	C2	D2	E2
3	A3	B3	C3	D3	E3
4	A4	B4	C4	D4	E4

Fig .1 IPC grid architecture

Every node in the grid architecture is an independent process and should keep sending data to adjacent nodes. Also, every node acts as a reference node that receives data from neighbors for detecting whether an event occurs, the nearest neighbour communication scheme is shown in Figure 2. The node should report the event and information recorded to the base station when it receives at least three same values from adjacent nodes at the same time. To make sure the efficiency of inter-process communication, the IPC scheme chosen should minimize the messages to the base station when an event occurs.

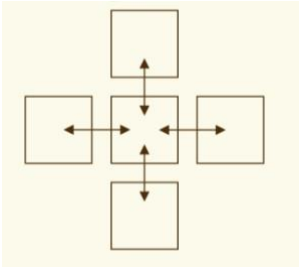

Figure. 2 nearest neighbour communication

B. Design of IPC architecture

The pyramid network is one of the most important network topologies as it is popular in both hardware and software structures for parallel computing [3]. The pyramid network model is good at representing the WSN network, in terms of Figure 3 below, the top point can be represented as the base station and points in the bottom can be seen as nodes. The advantages of the pyramid network are it can support nearest neighbour communication for local communication of 20 nodes and supports a tree communication for global communication between the base station and nodes.

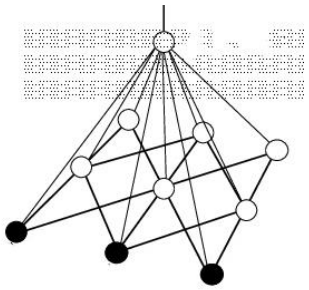

Fig. 3 Pyramid network

To achieve the communication scheme between nodes and the base station (tree communication), the master slave technology is chosen, which is most of the processes used to compute and one process managing all other processes, as shown in Figure 4.

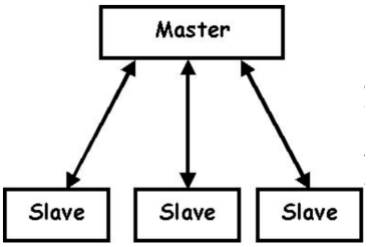

Fig. 4 Master slave

To achieve the nearest neighbour communication for 20 nodes, MPI Cartesian Topology is chosen as it can create a new communicator and mapping virtual topology onto the 4×5 grid architecture.

C. Implementation of IPC architecture

In the IPC architecture chosen, we first split the communicator to one communicator for the base station and one communicator for 20 nodes, which is a master slave program. An MPI Cartesian Topology is used in the communicator of 20 nodes, creating a 4×5 grid architecture for processes. The advantage of MPI Cartesian Topology is it is easy to simulate the WSN architecture and find every node's neighbor process. The "-2" will be displayed when a process does not have a neighbor.

To observe events easily, a sliding window trick is implemented to increase the number of the event occurs. The window size in our experiment is 3. During each iteration, a matrix will store each reference node's adjacent nodes' value in current iteration and previous two iteration's value. Now, if values in the current iteration also match values in the previous two iterations, values in the previous iteration also are considered. For example, in Table 1 below, for this reference node, the adjacent node 0 and 3 have value 9 in the current iteration, a value 9 also occurs in adjacent node 2 in iteration 9. Thus, an event occurs and the activation value is 9 in iteration 10.

TABLE I

Adjacent nodes of one reference node	**Iteration=8**	**Iteration=9**	**Iteration=10 (current iteration)**
0	1	0	9
1	6	2	0
2	2	9	8
3	3	1	9

When an event occurs, the event and relevant information are sent to the base station using an EVENT tag. Events will happen during some iterations; the base station needs to keep receiving events for any source and any tag. When a node finishes its job, an EXIT tag will be sent to the base station. The base station will terminate to receive events when it receives 20 EXIT tags, knowing all nodes finish their job.

To achieve modularity, making the program is similar to actual WSN, the program is divided into 3 majority functions, which are the main function, base station function, node function. Figure 5, 6 and 7 illustrate the technical flowchart for the IPC scheme, making the scheme easy to understand.

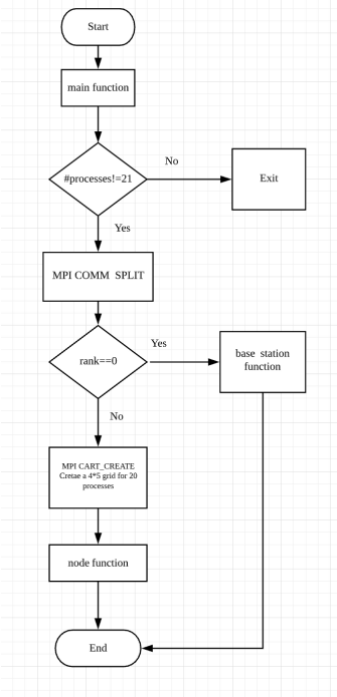

Fig. 5 main function

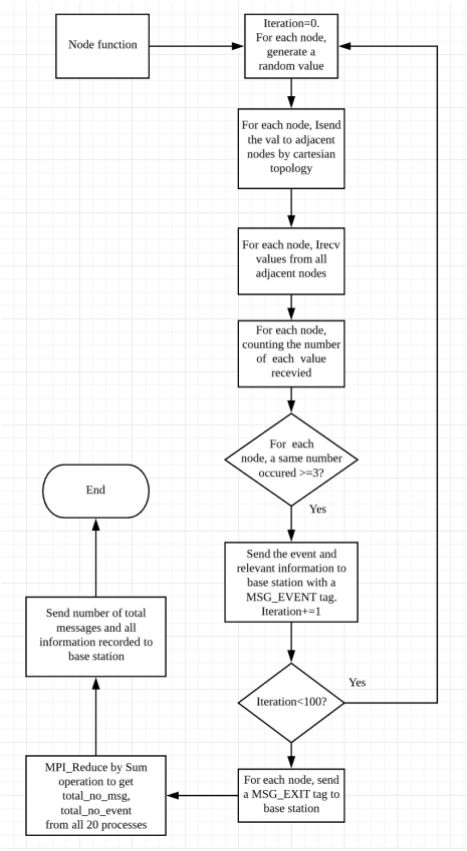

Fig. 6 node function

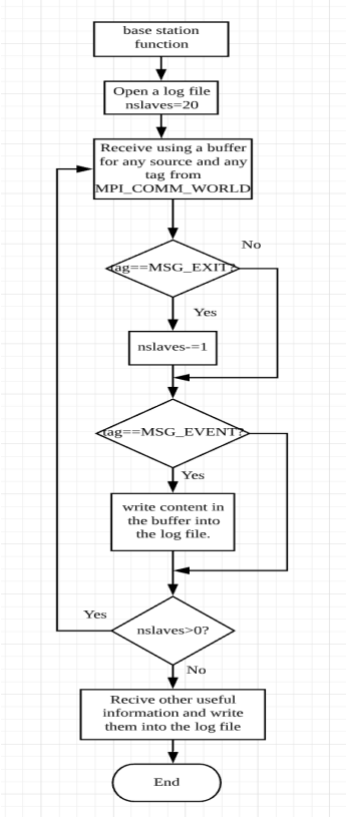

Fig.7 base station function

III. DESIGN OF ENCRYPTION ALGORITHM

The security between communication in WSN is important, there are two kinds of main communication need to be encrypted, one is the communication between adjacent nodes, which a message is only a number, the other one is the communication between nodes and base station, which is the event and detail of nodes, the length of message is longer.

AES is a complex algorithm that requires a large number of mathematical communications o be done [4]. It is based on the Substitution Permutation network. AES encrypts data by dividing data into a different data block, the basic unit processing is a byte. Each block in the AES encryption is transformed by a number of repetitions of rounds. For each round, there are several processing steps.

1. The plaintext is separated into blocks of size 128 bits, it separates text into a matrix.
2. Key expansion: Get Round keys by Rijndael key scheme.
3. Add round key: Key is added into the block of the message using XOR encryption.

4. Substitute bytes: each byte is substituted by using a predetermined table.
5. Shift rows: Rows are shifted accordingly.
6. Mix columns: A mixing operation applied to it to diffuse columns.
7. Add round key

After all rounds, the information in the final matrix is the encryption text. To decrypt the encryption text is easy by applying the inverse of each encryption step above. Figure 8 illustrates the technical flowchart for the AES encryption and decryption algorithm.

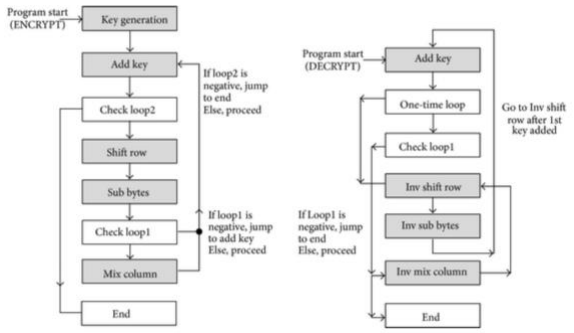

Fig .8 AES encryption & decryption

The AES algorithm is complex and spends much time to execute, so a parallel AES algorithm using OpenMP is implemented for improving the performance, which can be used to perform shared memory parallelism. The approach for implementing the parallel AES algorithm is dividing each block into the independent task and dynamic scheduling tasks when any of the threads are free. The code implemented is modified from the link [5].

IV. RESULTS AND DISCUSSIANS

A. Sliding window Trick

The experiment result of simulating WSN is recorded in a log file, the simulation is run for 50 iterations, the total number of messages passed throughout the network is 3570 and the number of events occurs throughout the network is 449, as shown in Figure 9, the same simulation without sliding window is shown in Figure 10, we can observe the number of events occur has increased successfully. Also, a matrix that used to track information for every iteration is recorded in screenshot 2 and Screenshot 3 in appendices, it clearly shows that the number of events occurred in each iteration has increased. Thus, the sliding window trick has been successfully achieved.

```
Simulation time is  2.548453 s
Number of messages passed throughout the network:3570
Number of events occurred throughout the network449
```

Fig .9 WSN using sliding window

```
Simulation time is  2.351588 s
Number of messages passed throughout the network:3381
Number of events occurred throughout the network260
```

Fig .10 WSN without sliding window

B. Efficent Communication and Event Detection Criterion

A small extract of the log file is shown in Screenshot 1 in appendices, we can observe that every event's detail is recorded in the log file. For each event, the detail includes the date and time, the number of iteration, the reference node with mac address, the activation value reported and its adjacent nodes' rank number (-2 indicates no neighbour). That agrees with the WSN's event detection criterion. The communication time between nodes and the base station is also recorded, we can find that the time of each communication is very small, which indicates the communication is efficient. Thus, we have proved hypothesis 1, which is the scheme chosen can be implemented efficiently and satisfies WSN's event detection criterion.

C. Minimize the Number Of Messages Reported

The scheme only sends one message when an event occurs just like the message in the log file. Therefore, the number of messages per event being reported to the base station is only one, that has proved our hypothesis1, the messages per event reported to the base station is less than 4.

D. Parallel encryption

AES encryption algorithm is tested and the experiment in term of comparison between parallel implementation and serial implementation is recorded in Table II (A more specific experiment result is recorded in the Screenshot 4and Screenshot 5 in appendices). The parallel AES is parallelized by setting 8 threads and using a dynamic schedule. In term of Table II, the parallel encryption and decryption cost is only about $\frac{1}{4}$ time of serial implementation. Also, by observing the Screenshot 6, it clearly shows the AES encryption algorithm has successful encrypt and decrypt the message "hello world". The AES has been successfully implemented and has a speedup greater than 1, which completes the hypothesis3 presented.

TABLE II

Times of experiment	Serial Encryption time	Parallel encryption time	Serial decryption time	Parallel Decryption time
1	0.0359832	0.008224	0.078332	0.020466
2	0.031158	0.007773	0.076565	0.019211
3	0.030759	0.007942	0.078247	0.019898
4	0.031330	0.007843	0.078297	0.019946
5	0.033731	0.007991	0.082179	0.020153

V. CONCLUSIONS

The report describes an IPC architecture scheme. By reviewing some papers, a pyramid network is chosen for the nearest neighbor topology. We implement the scheme using master slave technology and MPI Cartesian Topology. Then, considering the security of communication, a parallel AES encryption algorithm is used to efficient encrypt the messages of communication. From the results of the experiment, we can draw the conclusion that the IPC scheme chosen can simulate the WSN efficiently and minimize the number of messages per event reported to the base station whilst satisfies the WSN' event detection criterion. Furthermore, the speed up improved is greater than 1 using OpenMP, which proves the parallelism of AES encryption algorithm can improve the performance.

There exist limits in the experiment. (1) All experiments are only run on a single multicore computer, the future plan is to further test the scheme on different hardware, getting some more generalized results. (2) Only one IPC scheme is tested in our experiment. In future, more investigation will be done and other schemes will be tested and be compared with the IPC scheme chosen currently.

REFERENCES

[1] "Wireless sensor network", *En.wikipedia.org*, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Wireless_sensor_network. [Accessed: 15- Oct- 2019].

[2] D. Pandey, "Inter Process Communication (IPC) - GeeksforGeeks", GeeksforGeeks, 2019. [Online]. Available: https://www.geeksforgeeks.org/inter-process-communication-ipc/. [Accessed: 15- Oct- 2019].

[3] M. Farahabady and H. Sarbazi-Azad, "The Grid-Pyramid: A Generalized Pyramid Network", *Dl.acm.org*, 2019. [Online]. Available: https://dl.acm.org/citation.cfm?id=1145455. [Accessed: 16- Oct- 2019].

[4] National Inst. of Standards and Technology, Federal Information Processing Standard Publication 197, The Advanced Encryption Standard, Nov 2001

[5] https://github.com/NihalHarish/ParallelAES-Encryption

APPENDIX

```
Thu Oct 17 17:08:18 2019: Iteration: 5 activation value:7 Reference Node:5[IP address:1.0.0.0 ]Adjacent nodes: 1 9 4 6
communication time between nodes and base station reported activation value is 0.000007 s

Thu Oct 17 17:08:18 2019: Iteration: 5 activation value:5 Reference Node:13[IP address:1.0.0.0 ]Adjacent nodes: 9 17 12 14
communication time between nodes and base station reported activation value is 0.000008 s

Thu Oct 17 17:08:18 2019: Iteration: 6 activation value:2 Reference Node:17[IP address:1.0.0.0 ]Adjacent nodes: 13 -2 16 18
communication time between nodes and base station reported activation value is 0.000007 s

Thu Oct 17 17:08:18 2019: Iteration: 6 activation value:5 Reference Node:17[IP address:1.0.0.0 ]Adjacent nodes: 13 -2 16 18
communication time between nodes and base station reported activation value is 0.000011 s

Thu Oct 17 17:08:18 2019: Iteration: 5 activation value:5 Reference Node:14[IP address:1.0.0.0 ]Adjacent nodes: 10 18 13 15
communication time between nodes and base station reported activation value is 0.000006 s

Thu Oct 17 17:08:18 2019: Iteration: 5 activation value:7 Reference Node:9[IP address:1.0.0.0 ]Adjacent nodes: 5 13 8 10
communication time between nodes and base station reported activation value is 0.000015 s

Thu Oct 17 17:08:18 2019: Iteration: 6 activation value:9 Reference Node:5[IP address:1.0.0.0 ]Adjacent nodes: 1 9 4 6
communication time between nodes and base station reported activation value is 0.000009 s

Thu Oct 17 17:08:18 2019: Iteration: 6 activation value:5 Reference Node:18[IP address:1.0.0.0 ]Adjacent nodes: 14 -2 17 19
communication time between nodes and base station reported activation value is 0.000012 s

Thu Oct 17 17:08:18 2019: Iteration: 6 activation value:2 Reference Node:13[IP address:1.0.0.0 ]Adjacent nodes: 9 17 12 14
communication time between nodes and base station reported activation value is 0.000007 s

Thu Oct 17 17:08:18 2019: Iteration: 6 activation value:3 Reference Node:13[IP address:1.0.0.0 ]Adjacent nodes: 9 17 12 14
communication time between nodes and base station reported activation value is 0.000010 s

Thu Oct 17 17:08:18 2019: Iteration: 6 activation value:5 Reference Node:15[IP address:1.0.0.0 ]Adjacent nodes: 11 19 14 -2
communication time between nodes and base station reported activation value is 0.000009 s
```

Screenshot 1. Small extract of the event reported associated with corresponding detail

Iteration	Number of msg passed	Number of events
0	78	16
1	70	8
2	74	12
3	72	10
4	71	9
5	68	6
6	73	11
7	72	10
8	69	7
9	72	10
10	71	9
11	71	9
12	69	7
13	71	9
14	71	9
15	70	8
16	68	6
17	71	9
18	68	6
19	71	9
20	70	8
21	70	8
22	72	10
23	68	6
24	70	8
25	71	9
26	70	8
27	75	13
28	70	8
29	72	10
30	66	4
31	71	9
32	73	11
33	72	10
34	74	12
35	68	6
36	73	11
37	74	12
38	72	10
39	72	10
40	71	9
41	74	12
42	71	9
43	71	9
44	69	7
45	72	10
46	71	9
47	73	11
48	70	8
49	69	7

Tracking matrix with sliding window(size=3)

Screenshot 2

Iteration	Number of msg passed	Number of events
0	78	16
1	73	11
2	71	9
3	68	6
4	64	2
5	63	1
6	64	2
7	64	2
8	64	2
9	65	3
10	67	5
11	66	4
12	66	4
13	65	3
14	68	6
15	68	6
16	64	2
17	65	3
18	65	3
19	66	4
20	69	7
21	70	8
22	68	6
23	69	7
24	66	4
25	66	4
26	66	4
27	65	3
28	64	2
29	67	5
30	68	6
31	66	4
32	67	5
33	66	4
34	68	6
35	67	5
36	67	5
37	67	5
38	69	7
39	71	9
40	72	10
41	69	7
42	68	6
43	68	6
44	67	5
45	66	4
46	67	5
47	68	6
48	67	5
49	68	6

Tracking martrix without sliding window

Screenshot 3

```
The experiment text is
 5
iteration 0
Serial encryption costs 0.037566
Serial decryption costs 0.077495
parallel encrption using OpenMP costs 0.008437
parallel decryption using OpenMP costs 0.022814

iteration 1
Serial encryption costs 0.032625
Serial decryption costs 0.077332
parallel encrption using OpenMP costs 0.008323
parallel decryption using OpenMP costs 0.021704

iteration 2
Serial encryption costs 0.031968
Serial decryption costs 0.080399
parallel encrption using OpenMP costs 0.008053
parallel decryption using OpenMP costs 0.022002

iteration 3
Serial encryption costs 0.030324
Serial decryption costs 0.075939
parallel encrption using OpenMP costs 0.008768
parallel decryption using OpenMP costs 0.021984

iteration 4
Serial encryption costs 0.030763
Serial decryption costs 0.078800
parallel encrption using OpenMP costs 0.008070
parallel decryption using OpenMP costs 0.021698

iteration 5
Serial encryption costs 0.030326
Serial decryption costs 0.075648
parallel encrption using OpenMP costs 0.008089
parallel decryption using OpenMP costs 0.021787

iteration 6
Serial encryption costs 0.030514
Serial decryption costs 0.076121
parallel encrption using OpenMP costs 0.008247
parallel decryption using OpenMP costs 0.021742

iteration 7
Serial encryption costs 0.031185
Serial decryption costs 0.082654
parallel encrption using OpenMP costs 0.008646
parallel decryption using OpenMP costs 0.023947

iteration 8
Serial encryption costs 0.033542
Serial decryption costs 0.079152
parallel encrption using OpenMP costs 0.008285
parallel decryption using OpenMP costs 0.022224

iteration 9
Serial encryption costs 0.030858
Serial decryption costs 0.076055
parallel encrption using OpenMP costs 0.008164
parallel decryption using OpenMP costs 0.022209
```

parallel encryption using length 1 message

Screenshot 4

```
The experiment text is
 Iteration: 8 activation value:5 Reference Node:1 Adjacent nodes: l r u d
iteration 0
Serial encryption costs 0.037238
Serial decryption costs 0.079946
parallel encrption using OpenMP costs 0.008305
parallel decryption using OpenMP costs 0.019888

iteration 1
Serial encryption costs 0.030862
Serial decryption costs 0.077187
parallel encrption using OpenMP costs 0.007850
parallel decryption using OpenMP costs 0.019302

iteration 2
Serial encryption costs 0.031198
Serial decryption costs 0.076151
parallel encrption using OpenMP costs 0.007738
parallel decryption using OpenMP costs 0.019395

iteration 3
Serial encryption costs 0.030971
Serial decryption costs 0.079757
parallel encrption using OpenMP costs 0.007997
parallel decryption using OpenMP costs 0.020770

iteration 4
Serial encryption costs 0.034872
Serial decryption costs 0.082245
parallel encrption using OpenMP costs 0.008055
parallel decryption using OpenMP costs 0.019420

iteration 5
Serial encryption costs 0.034103
Serial decryption costs 0.078165
parallel encrption using OpenMP costs 0.007714
parallel decryption using OpenMP costs 0.019486

iteration 6
Serial encryption costs 0.031327
Serial decryption costs 0.078042
parallel encrption using OpenMP costs 0.008013
parallel decryption using OpenMP costs 0.019683

iteration 7
Serial encryption costs 0.034927
Serial decryption costs 0.076690
parallel encrption using OpenMP costs 0.008142
parallel decryption using OpenMP costs 0.019609

iteration 8
Serial encryption costs 0.031556
Serial decryption costs 0.076186
parallel encrption using OpenMP costs 0.007764
parallel decryption using OpenMP costs 0.019559

iteration 9
Serial encryption costs 0.030787
Serial decryption costs 0.078154
parallel encrption using OpenMP costs 0.007739
parallel decryption using OpenMP costs 0.019364
```

parallel encryption using longer length message

Screenshot 5

```
The experiment text is
 hello world

The encryption content is ÏD Éé²)IÀLŷ%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%
âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç
%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã
¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}
ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç%âyA_²}ÀÝ■Yã¥ç

The decryption content is hello world
```

messages before, after encryption and after decryption

Screenshot 6