SCHOOL OF
ELECTRICAL AND
ELECTRONIC ENGINEERING

THE UNIVERSITY of ADELAIDE

Is secure communication possible?
Final Seminar

Yuanhao Liu, Christopher Lau

Supervisors

Prof. Derek Abbott

Dr. James Chappell

Mr. Lachlan Gunn

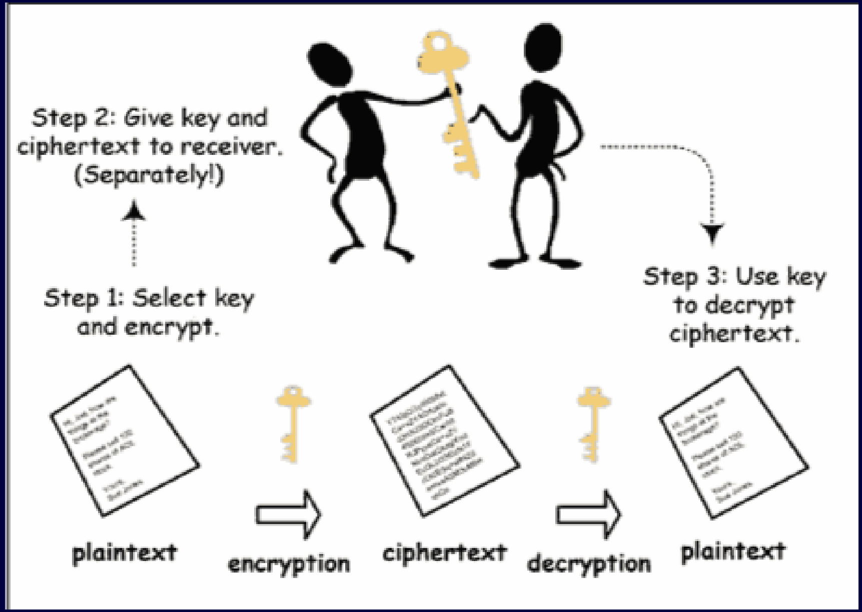
Outline

- Introduction
  - Project Aims
  - One Time Pad
- Design
  - Round Trip Times
  - Parity Iteration Protocol
  - Cascade Protocol
- Implementation
  - Socket Programming
  - Eavesdropper
- Project Management
- Conclusions and Questions

Introduction

Aim

- Investigation of Timing Based Key Agreement:
  - Its operation
  - Its limitations
- This method of generating a secure key is proposed to be a classical alternative to quantum key distribution



One Time Pad

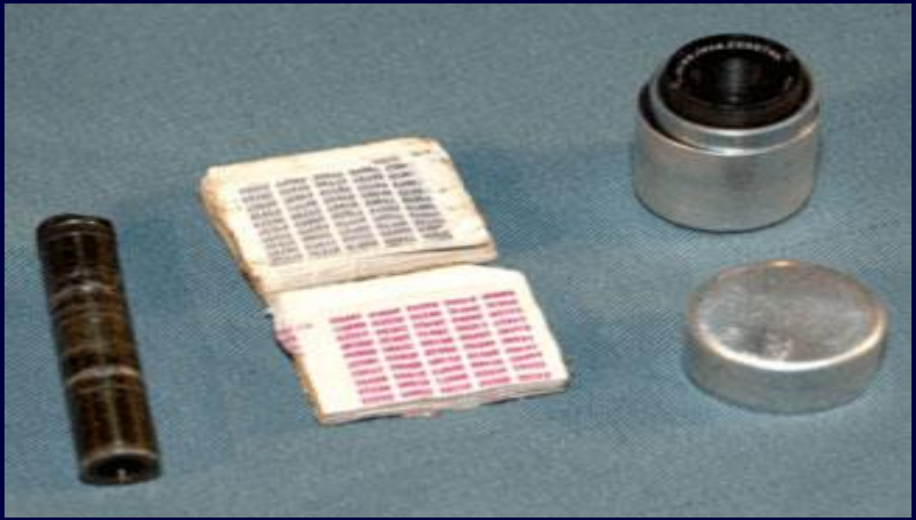
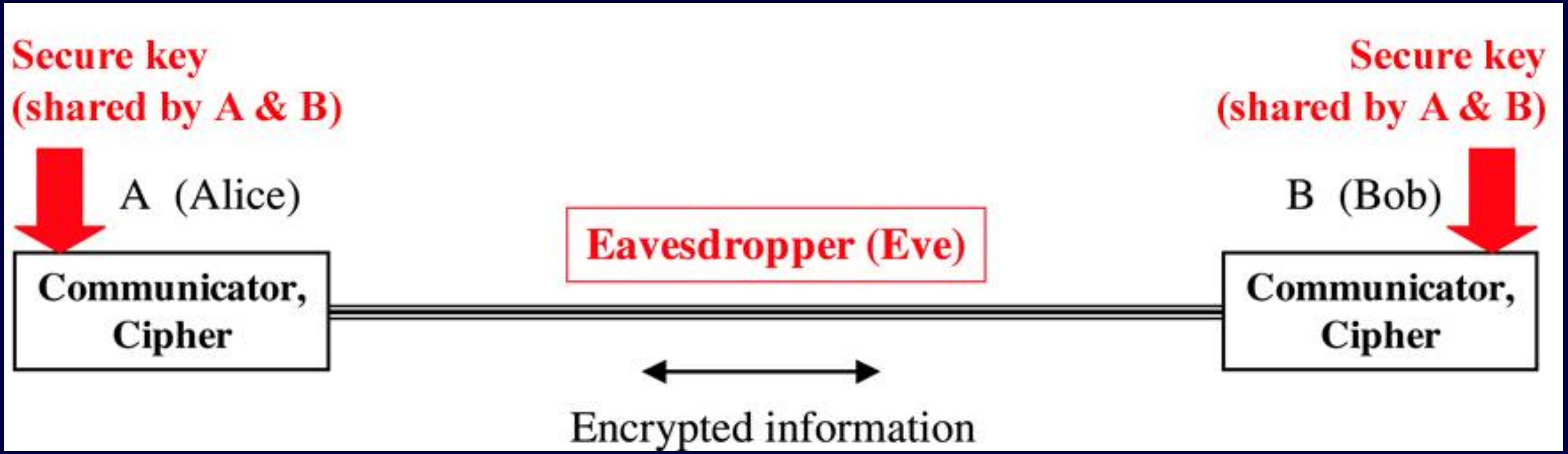


Figure: Private key transmission [4]

One Time Pad


Secure key
(shared by A & B)
A (Alice)
Communicator,
Cipher
Eavesdropper (Eve)
Encrypted information
Secure key
(shared by A & B)
B (Bob)
Communicator,
Cipher

Round Trip times

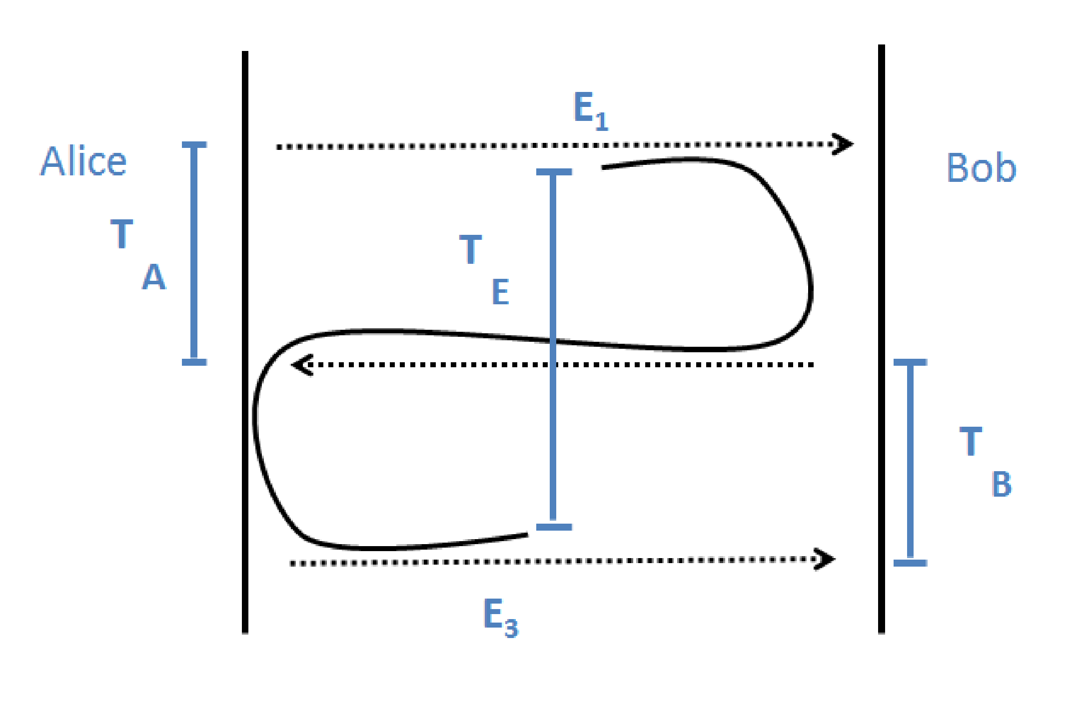
Round Trip Time for Alice: T_A
Round Trip Time for Bob: T_B
Round Trip Time for Eve: T_E

Ideal Case

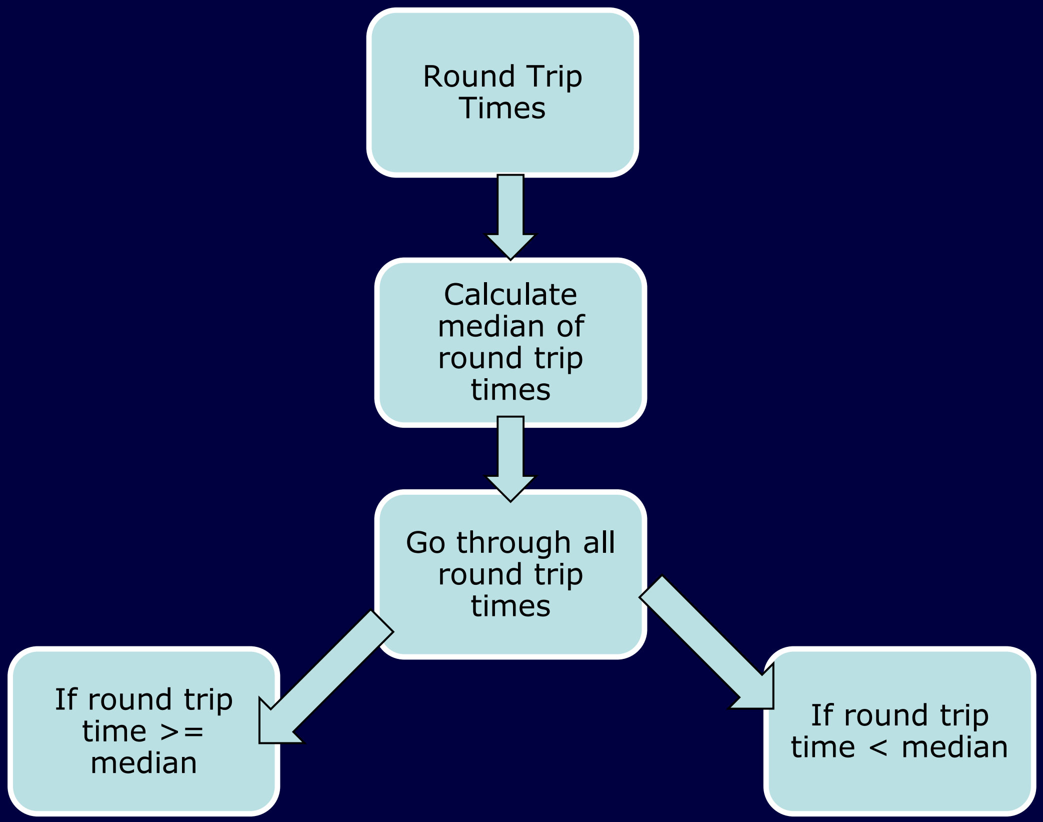
$T_A = T_B != T_E$

Real Case

$T_A != T_B != T_E$



Round Trip Times to Bit Stream



Output: 1 0

Parity bit generation

Bit A	Bit B	Parity bit
1	1	0
1	0	1
0	1	1
0	0	0

Parity iteration protocol

Alice: 0101 1010 1001 1001 1101 1101 1100

Bob: 0101 1011 0110 1010 1111 0011 1010

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

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

Alice: 0 0 1 1 0 1 0 1 1

Bob: 0 0 1 0 1 1 1 1 0

Parity iteration protocol (example)

BER: Bob = 0.2; BER: Eve = 0.1

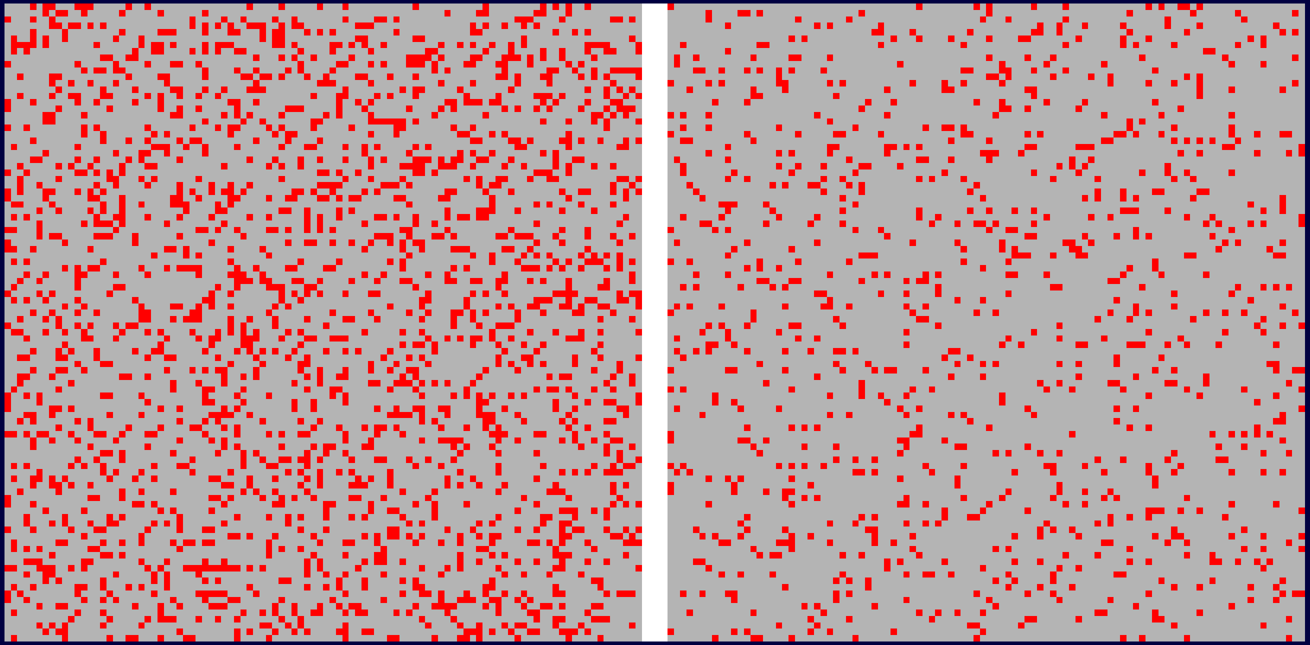


Figure: Left: Bob's errors, Right: Eve's errors. [3]

Parity iteration protocol (example)

After one iteration

BER: Bob = 0.06; BER: Eve = 0.1

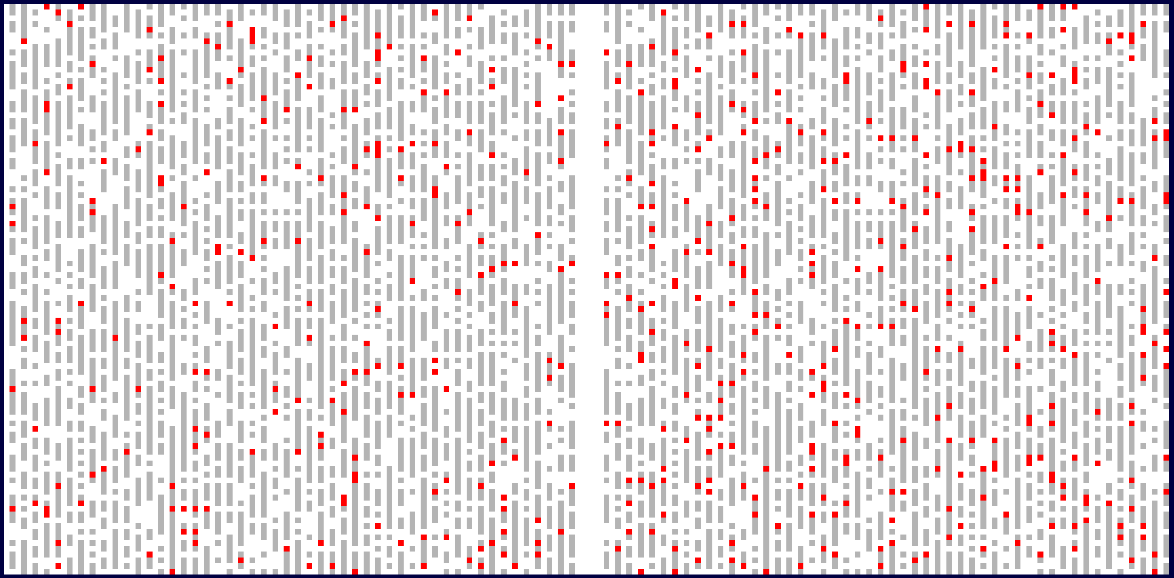


Figure: Left: Bob's errors, Right: Eve's errors. [3]

Parity iteration protocol (example)

After two iterations.

BER: Bob = 0.005; BER: Eve = 0.1

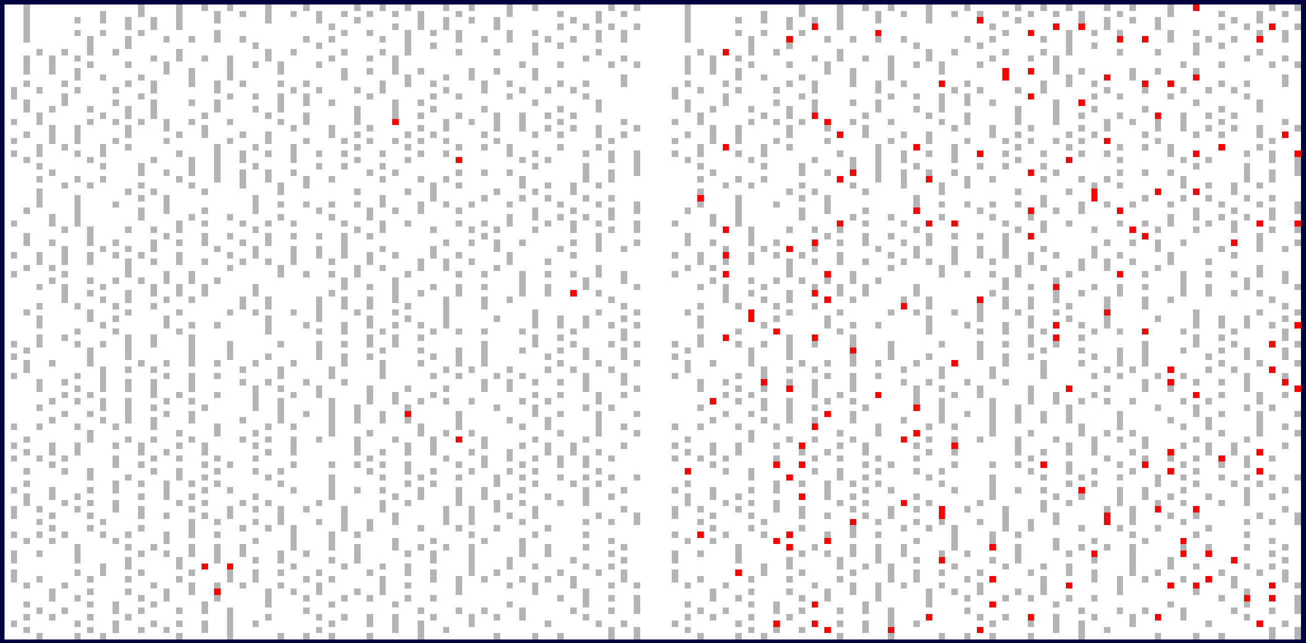


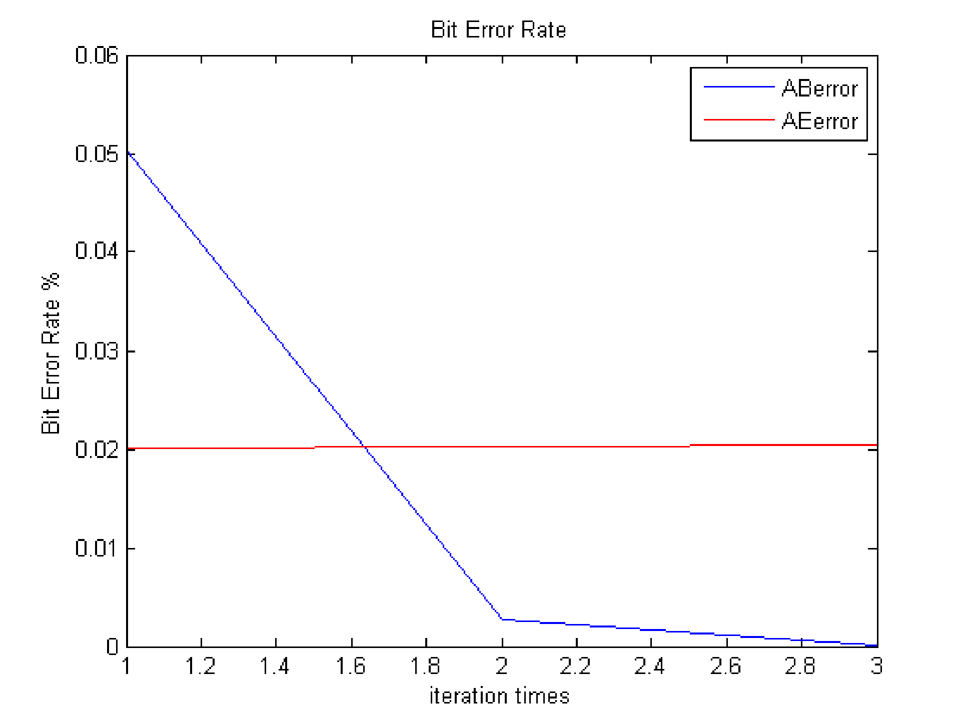
Figure: Left: Bob's errors, Right: Eve's errors. [3].

Parity iteration protocol

•Parity bits of each block are computed

•Parity bits of corresponding blocks are compared

•Retain the blocks for which the corresponding parity bits are the same and remove the blocks for which the corresponding parity bits are different



Cascade Protocol

Compute parities of each half of block

If left halves have same parity bits

Check within right half of block

If left halves have different parity bits

Check within left half of block

Alice: 01010110   0   0
Bob: 10010010   0   1
✓   x

Alice: 0110   1   1
Bob: 0010   0   1
x   ✓

Alice: 01
Bob: 00

Sixth bit is incorrect

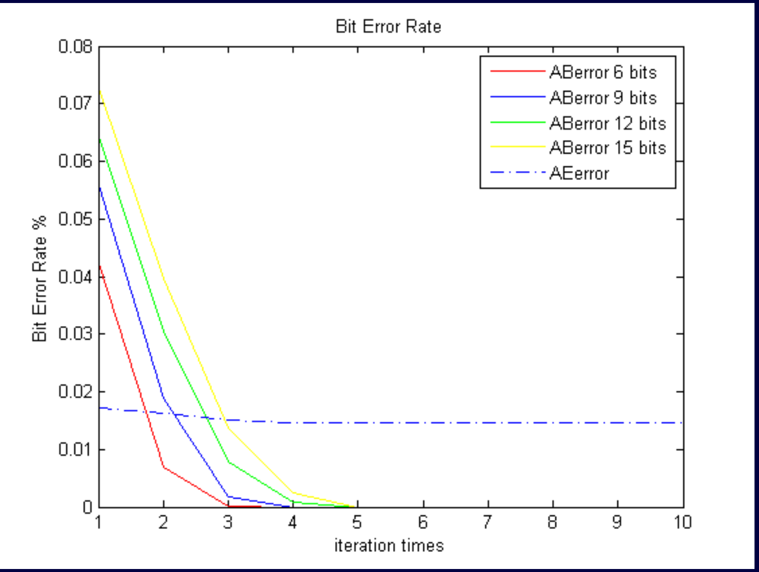
Cascade Protocol

Alice: 01010110
Bob: 10010010

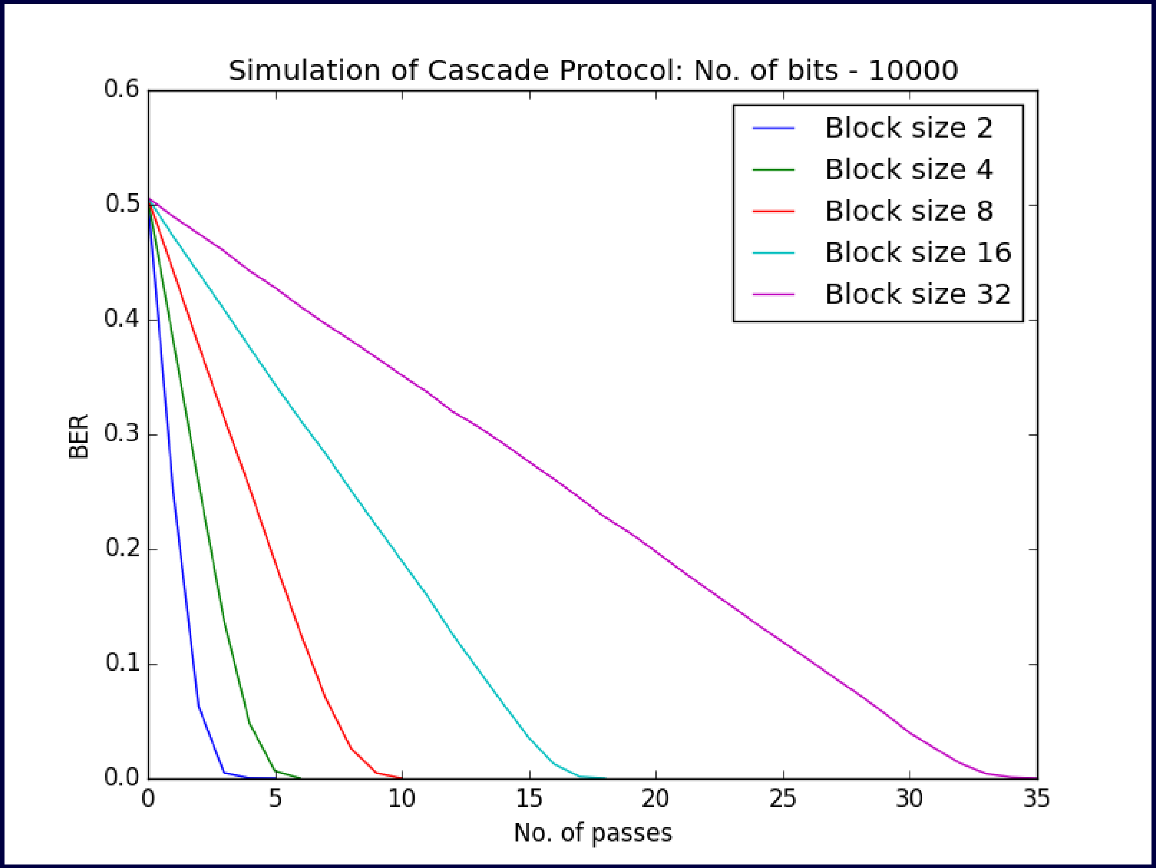
Cascade protocol

Alice: 01010110
Bob: 10010110

- Correct the error bit in each block
- Entire bitstream is randomly shuffled after correction
- Keep correcting error until bit error rate is within an acceptable level



Cascade Protocol



Comparison of data transmitted in each protocol

Cascade Protocol	Parity Protocol
$\text{data} \cong n * m * \log_2 k$ n: passes m:block numbers k: block sizes	$data \cong z$ z:string length

Cascade Protocol

Limitations

- Trade-off between block size and information leakage
- For each pass (going through all the blocks), only one error can be corrected at a time
- The parity check does not always indicate the correctness of the corresponding blocks
- This is because if there was an **even** number of errors, the parity check would indicate that the parity of the corresponding blocks are the same despite the presence of errors

Socket Programming

Data transmitted

- •Parity bit of Alice
  “1” or “0”
- •Response from Bob
  “correct” or “error”
- •Random number

```
        SERVER: Waiting for incoming connection ...
        A connection was found!
1 is sent
message from client: error
alice: 11010101
alice: 1101
alice: 01
alice: 0

        parity_b: 0
CLIENT: Do you want to connect to this server? (Y/N)
y
Reply received

message from server: 1
Data Send

bob: 00111111
Reply received

message from server the parity bit: 1
parity_left_b: 0
bob: 0011
Reply received

message from server the parity bit: 0
parity_left_b: 0
bob: 11
Reply received

message from server the parity bit: 0
parity_left_b:1
bob: 1
location: 2
bob: 00011111
```

Socket Programming

```
        SERVER: Waiting for incoming connection ...
        A connection was found!
0 is sent
message from client: correct

        parity_b: 0
CLIENT: Do you want to connect to this server? (Y/N)
y
Reply received

message from server: 0
Data Send
```

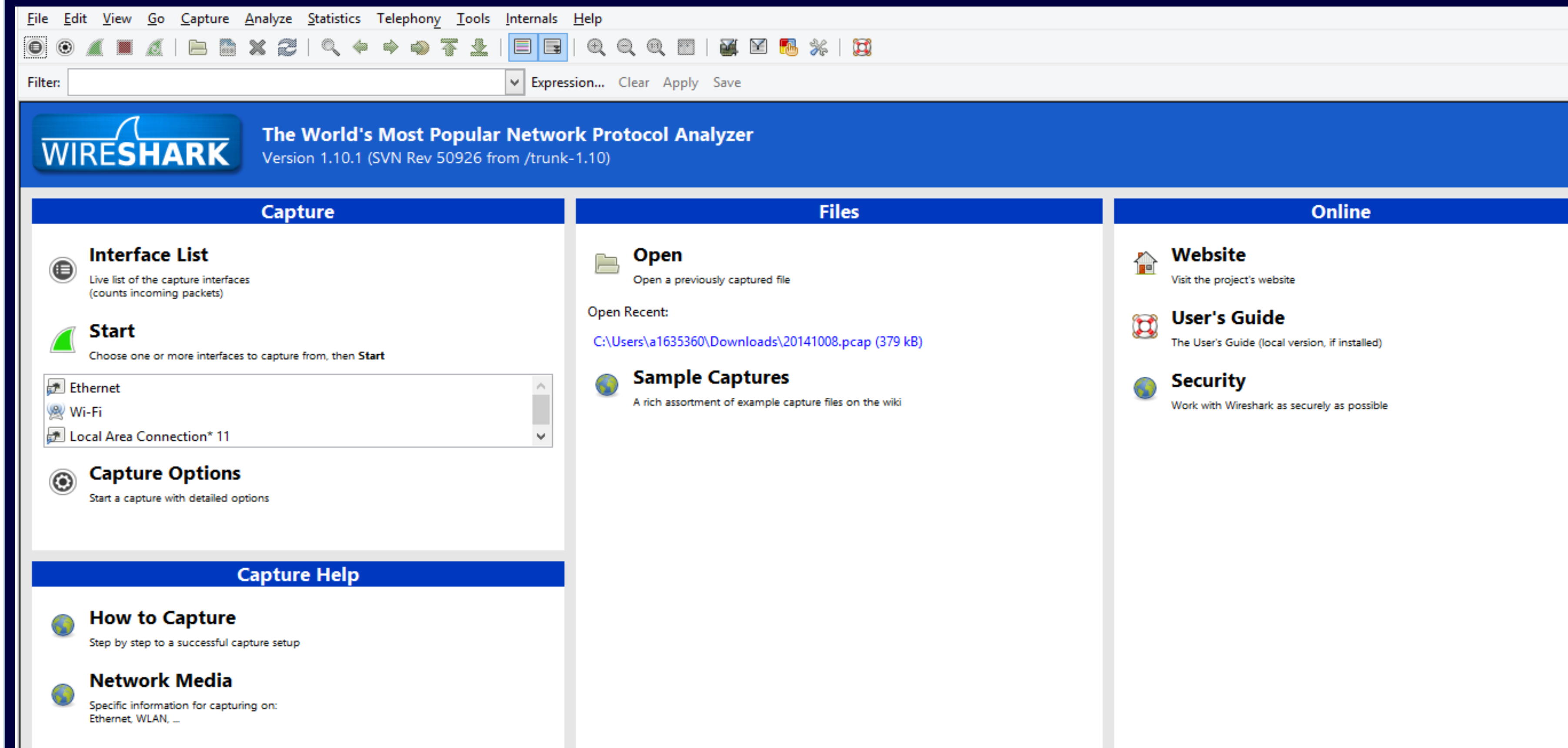
Socket Programming

```
        SERVER: Waiting for incoming connection ...
        A connection was found!
0 is sent
message from client: correct
begin to shuffle
message from client: 120Massage send to client: 120
CLIENT: Do you want to connect to this server? (Y/N)
y
Reply received

message from server: 0
Data Send

begin to shuffle
message from server: 120Massage send to server: 120
```

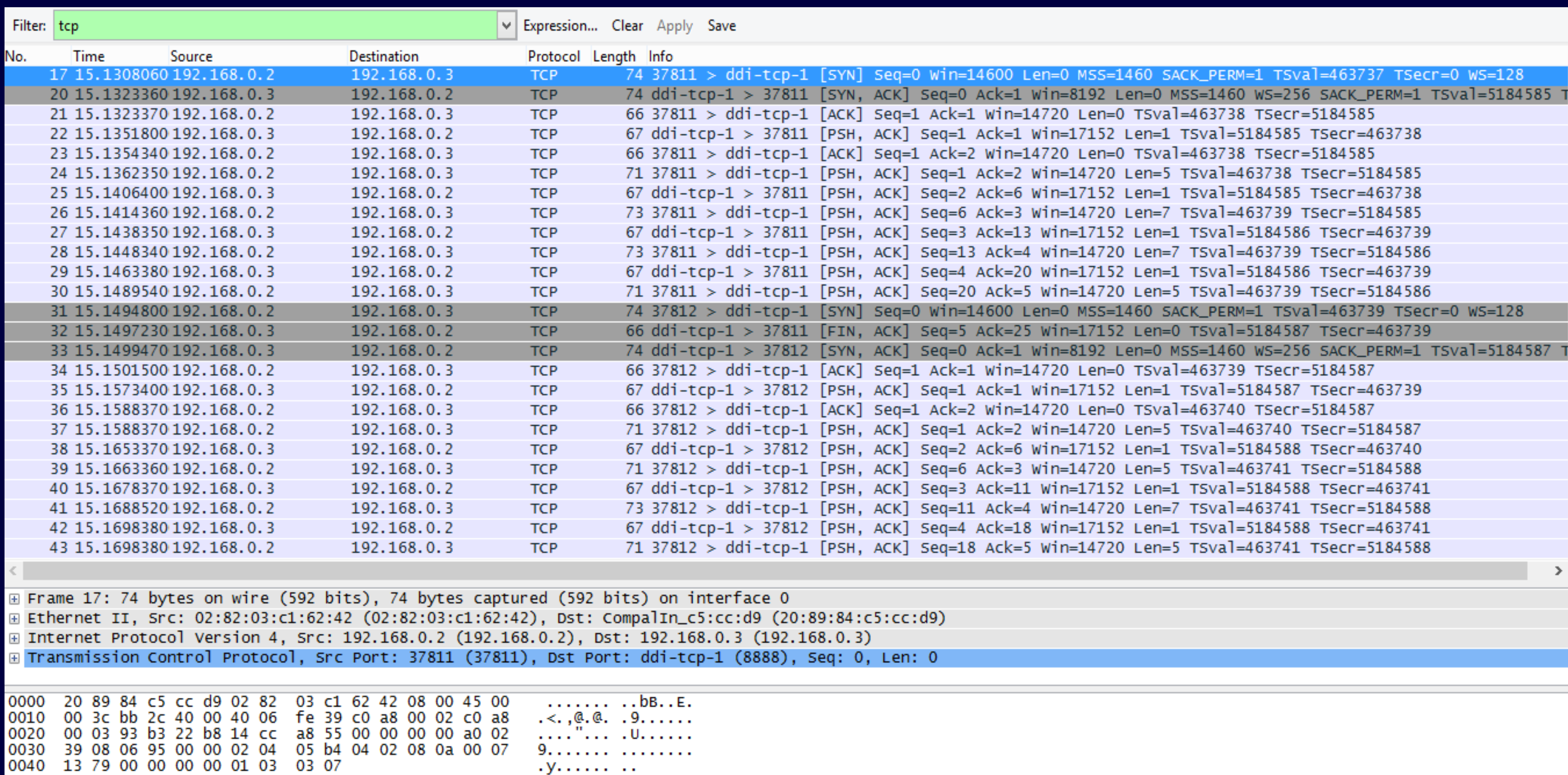
Eavesdropper

- The only info that the eavesdropper is able to obtain is the parity bits.
- She relies on these bits to determine whether which of her bits needs to be corrected
- Therefore, if Eve solely relies on the parity bits to correct her bit stream, her bit error rate is expected to not decrease as much as between Alice and Bob

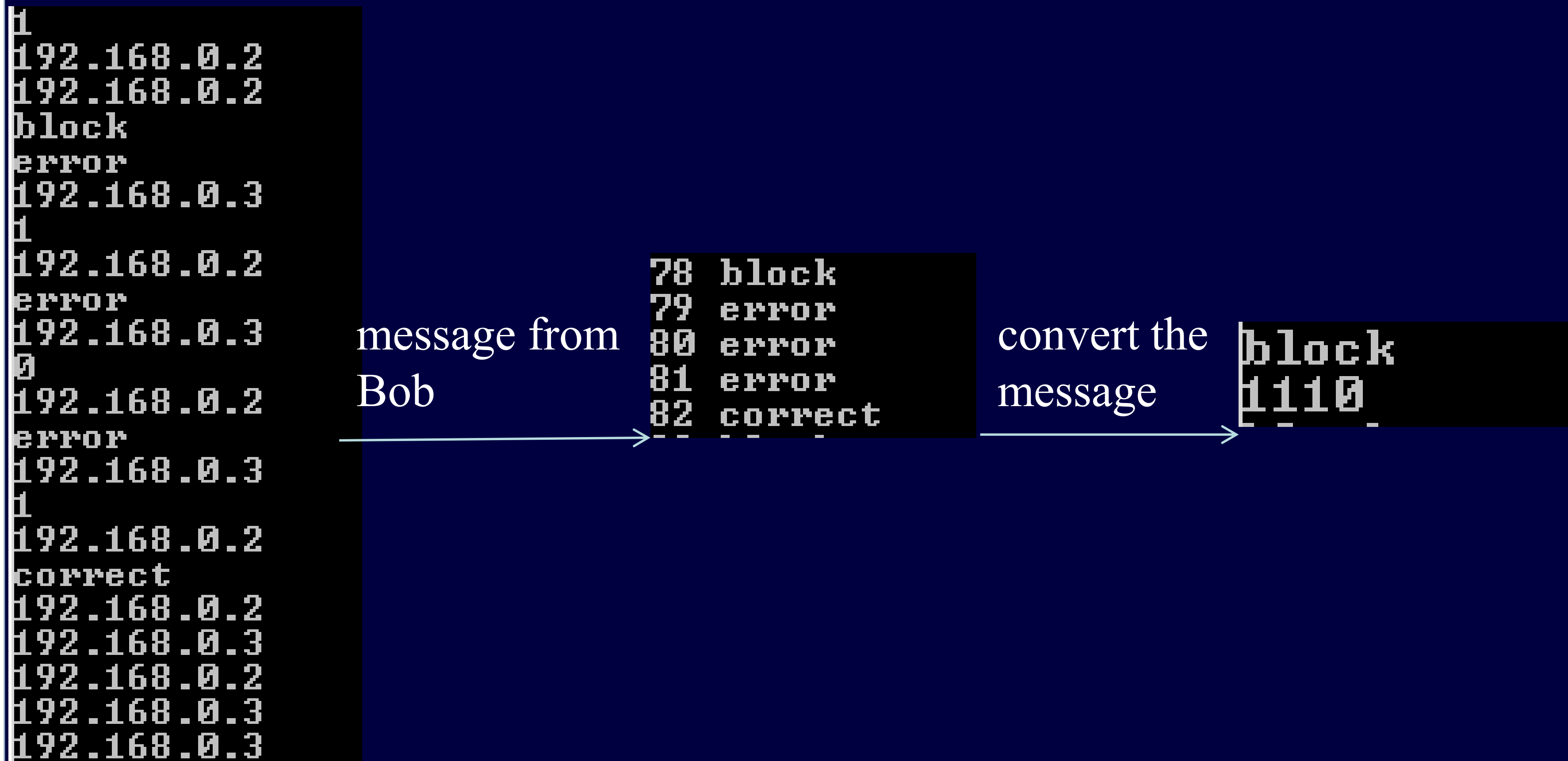
Monitoring the network transmission



Packets transmitted



Data Analysis

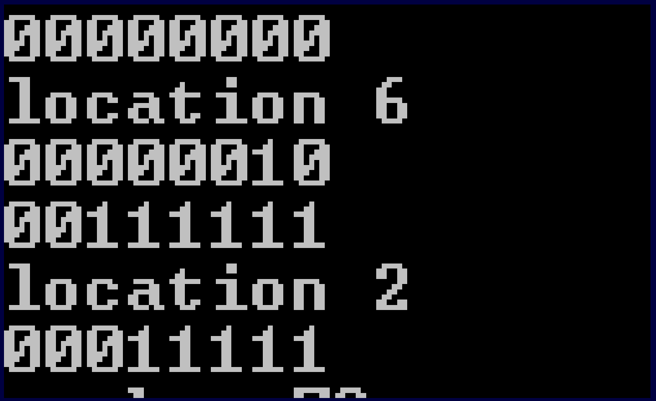


Cascade protocol for Eve

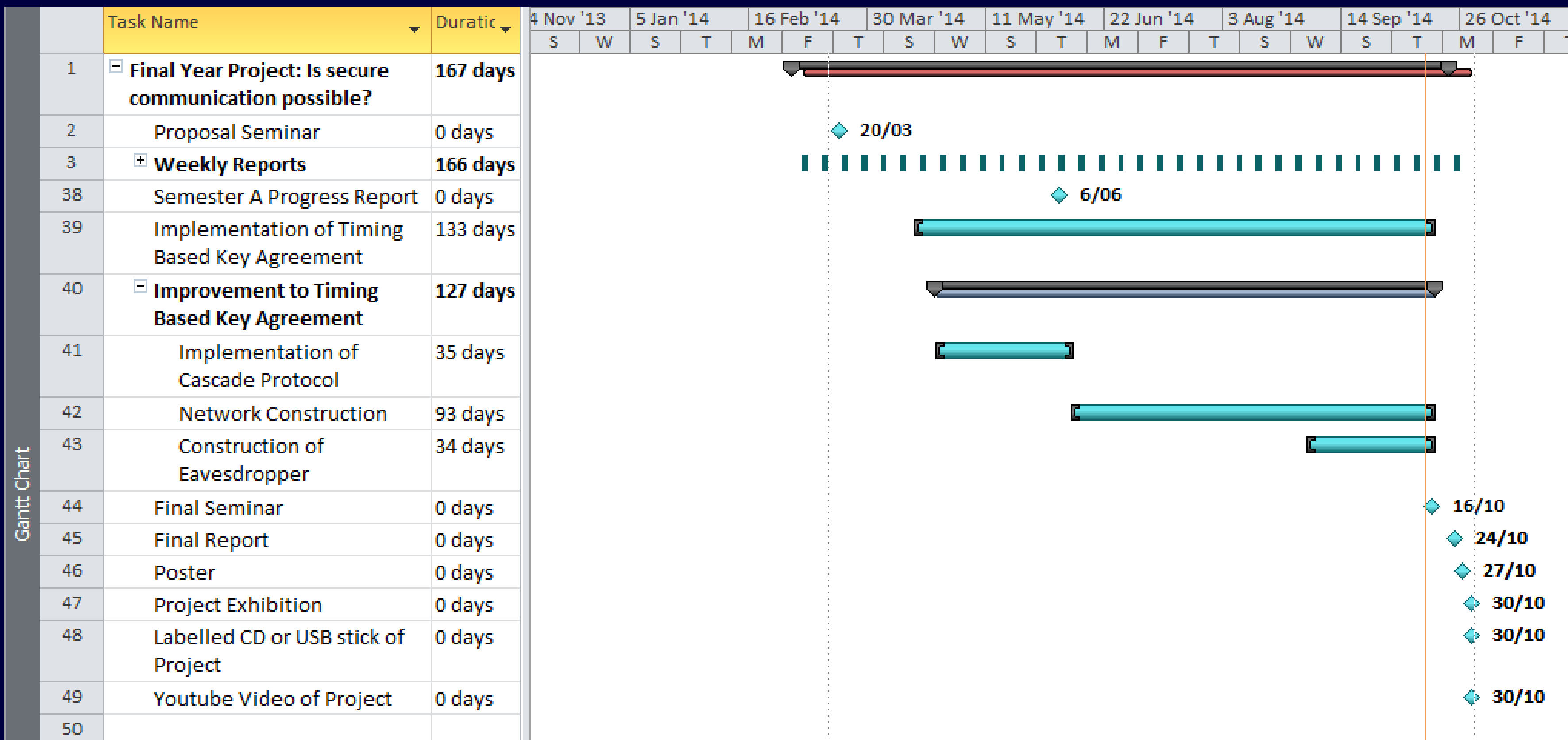
Original block bits

Location of error

Block bits after correction



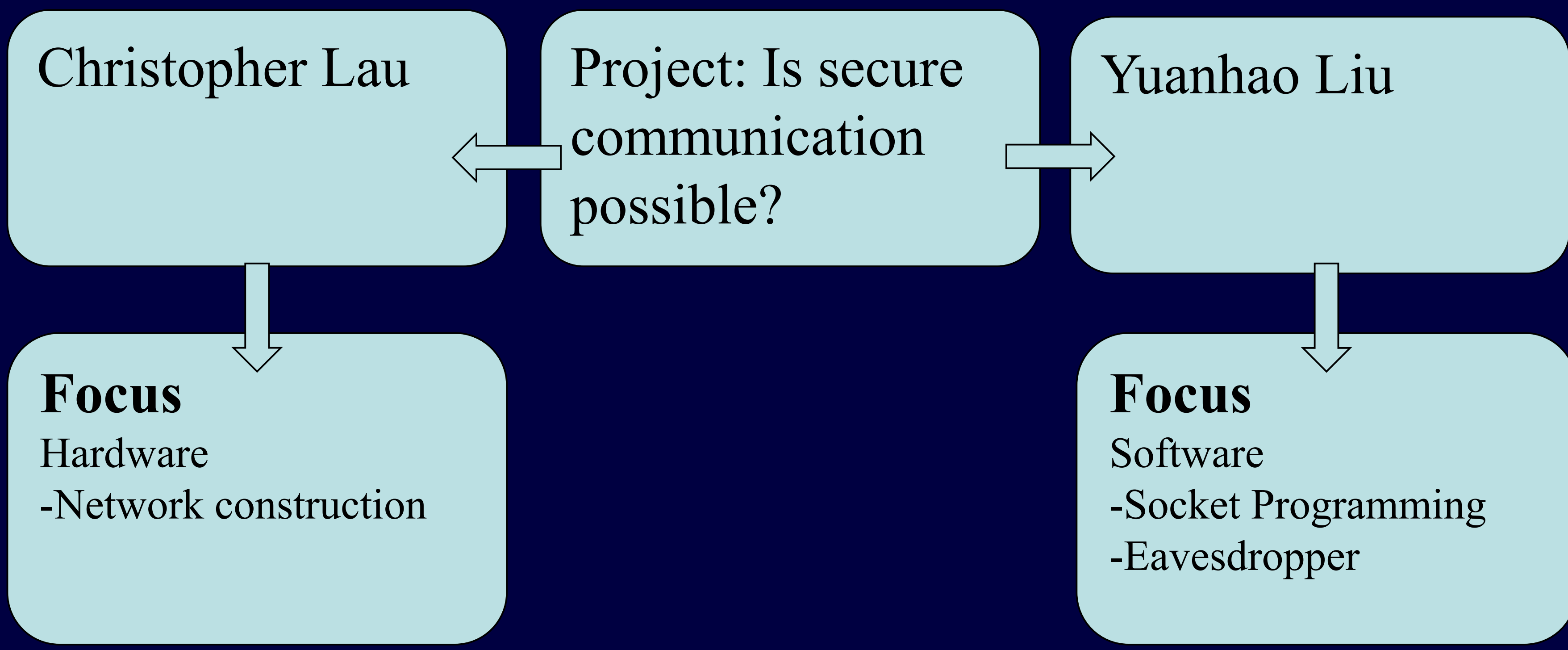
Gantt Chart


Gantt Chart
Task Name
Duratic
4 Nov '13
5 Jan '14
16 Feb '14
30 Mar '14
11 May '14
22 Jun '14
3 Aug '14
14 Sep '14
26 Oct '14
S W S T M F T S W S T M F T S W S T M F T
1 Final Year Project: Is secure communication possible? 167 days
2 Proposal Seminar 0 days
3 Weekly Reports 166 days
38 Semester A Progress Report 0 days
39 Implementation of Timing Based Key Agreement 133 days
40 Improvement to Timing Based Key Agreement 127 days
41 Implementation of Cascade Protocol 35 days
42 Network Construction 93 days
43 Construction of Eavesdropper 34 days
44 Final Seminar 0 days
45 Final Report 0 days
46 Poster 0 days
47 Project Exhibition 0 days
48 Labelled CD or USB stick of Project 0 days
49 Youtube Video of Project 0 days
50
20/03
6/06
16/10
24/10
27/10
30/10
30/10
30/10

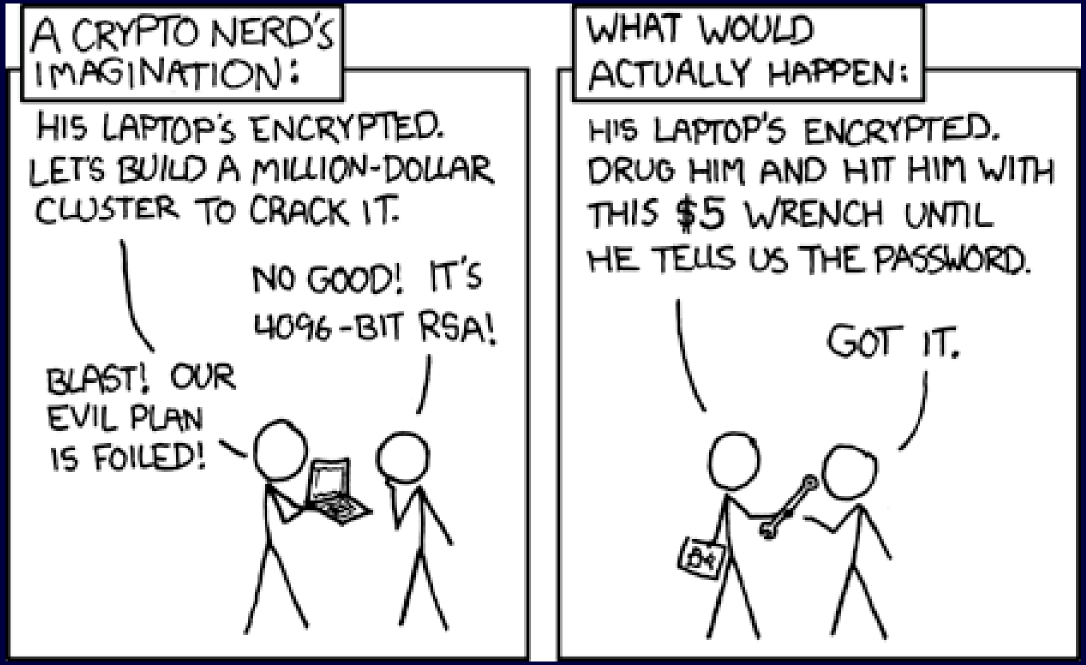
Risk Management

Risk	Likelihood	Severity	Avoidance / Mitigation Strategies
Project files missing or are not accessible	Very Low	Low	We will ensure that all work related to the project is stored on the cloud via the project's Wiki, Google Drive and Dropbox and is accessible by both team members.
Unavailability of team member	Low	Medium	Team members will keep each other informed about their work on the project which will allow their work to continue should a team member be unavailable.
Physical parts do not arrive and/or do not work in time for project completion	High	Medium	Ensure constant communication is sought with the suppliers regarding the progress of the delivery in order to plan for contingencies which include expediting work on improvements to Timing Based Encryption
Falling behind schedule due to increased complexity of work undertaken	Medium	Medium	Revaluate the scope of the project and if necessary restrict the scope to focus, among other things, on improving the functionality of the Timing Based Encryption and the encryption's efficiency.
Not finding a solution to our project	Very High	Very Low	Ensure any progress made is documented and can be used as the basis for the final seminar and exhibition

Project Management



Conclusion


A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.

Questions?

References

[1] Mateo J. M. et al. 'Demystifying the Information Reconciliation Protocol Cascade', http://arxiv-web3.library.cornell.edu/pdf/1407.3257v1.pdf

[2] Brassard G., Salvail L. 'Secret-Key Reconciliation by Public Discussion' http://link.springer.com/content/pdf/10.1007%2F3-540-48285-7_35.pdf

[3] Lachlan J. G., James M. C., Andrew A., Derek A., 'Physical layer encryption on the public internet: a stochastic approach to the Kish – Sethuraman cipher', School of Electrical and Electronic Engineering, The University of Adelaide, 31/10/2013.

[4] Tradecraft Artefacts, Canadian Security Intelligence Service, https://www.csis.gc.ca/hstrrtfcts/rtfcts/trdrtfctsndx-en.php