E91 Secure Messaging

Valene Tjong, Franck Belemkoabga, Mond Ibrahim, Nikkin Dev

Meet the team.

Valene Tjong

Junior Physics and CS student at Cornell University,

Franck Belemkoabga

Sophomore EECS student at MIT

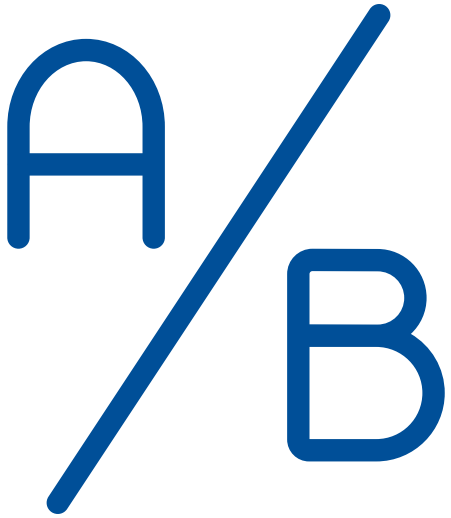
Mond Ibrahim

Computer Engineering PhD student at NC State; researches Quantum Computing Architectures and implementations QC Algorithms

Nikkin Dev

MSc. Physics -- Cologne, Germany; loves thinking about Quantum Algorithms in his spare time. (Has a cute dog)

Project Overview



Ekert-91 (E91)

Protocol utilizing two properties of entanglement generated by a common source, to distribute a sifted key to two parties Alice and Bob, with the presence of an eavesdropper Eve.

Project Overview

Text Encryption Module

AES (Advanced Encryption Standard) text encrypter and decrypter module using QKD key

Project Overview

Email and A-to-B Secure Chat CLI

1. Email module
2. Server-client chatting CLI

Project Overview

Website w/ Quantum Secured Chat

Secure chat room encrypted with AES and QKD key.

What is Ekert-91?

E91 is a protocol that utilizes two properties of entanglement--generated by a common source--to distribute a sifted key between two parties (Alice and Bob), in the presence of an eavesdropper Eve.

Why E91?

- The **entangled states** are either *perfectly correlated* or perfectly non-correlated, demonstrating **directionality synchronization** btwn parties
- Any eavesdropping attempts by Eve can be detected as they destroy the correlations between Alice and Bob.

E91.py usage

1. Select 3 bases for Alice, 3 bases for Bob, with 2 common bases between them
2. Create entangled bell state between Alice and Bob
3. Choose a direction for measurement randomly - one for Alice, one for Bob
4. Measure entangled state in these directions/basis
5. Share the basis used for different rounds
6. Use subset of measurements with different basis to test CHSH inequality
7. If the inequality satisfy quantum correlations, use subset with same basis to generate key

E

91

Project Demo

Checklist

Launch Timeline

Future plans for scaling our project.

Feb 22	Improve Email Interface	Improve Noise Algorithm	Link modules from front to end	Set up working examples
Mar 22	Add Texting Support (abuse hotline)	Improve modularization of packages		
Apr 22	Add support for 3 servers and clients for QKD and CLI	Continue improving quantum noise algorithm		
May 22	Scale to n servers and clients	Revisit literature for further innovations in QEC		
Soft Launch	Launch event			
After Launch	Follow up for testimonials	Process feedback	Plan for improvements	