On Secure Specifications for Large-Scale Quantum Key Distribution Implementations

Chase Kanipe and Adam Jeneski

Department of Physics, University of Maryland, College Park, MD 20742

Abstract. This paper addresses several security issues facing large scale or distributed implementations of the BB84 quantum key distribution protocol. Firstly, two simulations address how close the tolerated error and the actual error in the hardware must be. These results are quantified in the graphs below.

Keywords: Quantum Key Distribution, Security, Cryptography, BB84
PACS: 03.67.Dd

INTRODUCTION

Computer security is approaching a paradigm shift due to developments in quantum technologies. Many public key cryptosystems like RSA will be rendered obsolete in the coming years by quantum computers and Shor's Algorithm.[1] This major issue has led to many different developments in post-quantum cryptographic systems. One possible mitigation for the threat posed by quantum computers is the implementation of a quantum key distribution system.[2] While standard cryptographic schemes rely on the computational difficulty of solving certain mathematical problems (like the factoring problem for RSA), the security of quantum key distribution is ensured by quantum uncertainty of measuring certain systems.[3] This is preferable because continued increases in computational power can eventually render standard mathematical cryptographic schemes obsolete, while the security of quantum key distribution systems is not dependent on an attacker's computational power.[2]

This paper assesses the security of a large scale, photon polarization based implementation of the BB84 quantum key distribution (QKD) protocol.[3] The BB84 protocol attempts to leverage the quantum uncertainty of measuring photon polarization to ensure that a theoretical attacker, henceforth named Eve, cannot intercept the data without some statistical chance of altering the states in a detectable way.

Verified Bit Count and Hardware Noise

In the standard BB84 protocol implementation, for every photon measured by Eve, there is a 25% chance her measurement will be detectable to Alice (a sender) and Bob (a receiver), and a 50% chance she will be able to recover the correct bit.[3] The compounding of this detection chance over potentially thousands of bits in conjunction with privacy amplification algorithms ensures that Eve cannot intercept a significant portion of the data without being detected. However, a large-scale implementation of this protocol could face certain issues that a standard implementation does not. For example, if Eve had access to a high traffic central node, she could potentially have thousands or even millions of measurement attempts if she isn't targeting a specific victim. In this scenario, her key recovery rate could be much higher than the expected value, given a sufficient number of attempts. These statistics must be fully understood in order to identify the appropriate number of bits that the sender and receiver should compare in order to identify signs that the data has been intercepted.

Another important practical implementation issue to take into account is the error present in the hardware. To Alice and Bob a discrepancy caused by hardware uncertainty and a discrepancy caused by Eve performing an intercept-resend attack are indistinguishable.[3] Due to the inevitable lack of hardware reliability it is necessary that the protocol allows for a certain amount of error. If this error bound is too high, then it leaves room for Eve to intercept a certain percentage of the exchanged information, but if it is too low, it will result in a false positive.

I decided to address these issues using a series of simulations. The first step was to simulate the error variances in a hypothetical quantum channel. Greater error variance would lead to a greater range between the actual channel error and the error bounds allowed by the protocol. A second simulation is used to show what bit percentage is recoverable by Eve given a certain error range. Combining the data from these two simulations enabled me to identify constraints on a secure protocol.

Key Recovery Rate Simulation

To accomplish the first simulation, I created a Python program to show how the base hardware error rate and the verified bit length would affect the error variance, by directly stimulating photon measurement test cases with the relevant statistics from related literature. As would have been intuitively presupposed, greater base hardware error rate also led to greater error variance, while increasing the verified bit length helped mitigate this issue. Again, greater hardware error variance creates more space for Eve to potentially measure the photons undetected.

We then found the best fit plane for the data set visualized in Figure 1. This would enable us to estimate the potential error variance for any verified bit length and initial hardware error rate combination. It is important to emphasize that this simulation focuses on the feasible, statistical improbabilities. This is because in a distributed system Eve could have thousands or millions of attempts against different clients, and thus it is important to consider the worst case when assessing the security of the cryptographic system.

Recoverable Key Percentage Simulation

The second simulation shows how much of the key Eve could recover given a certain amount of allowed error. To do this I wrote a Python simulation for the BB84 protocol for a variety of error ranges and key lengths. To account for the distributed system scenario, I used the highest recovery percent given 1000 independent simulations. I found that the possible recovery rates could differ significantly from the expected values.

The results are visualized in Figure 2. Each line denotes the percent of key recovered (y-axis) given a certain key length and error rate. For example, the orange line corresponds to an 11% allowed error while the grey line corresponds to the 9% allowed error case. The chart shows the degree to which increasing the key length and decreasing the allowed error amount would decrease the key percentage recoverable by Eve.

As can be seen from the chart, while given a 10% allowed error, it is possible for Eve to recover around 30% of the exchanged information. This shows how while it is expected that Eve would recover 20% of the key information at a 10% detection rate,[2] in a distributed system with multiple attempts Eve could recover significantly more information.

Protocol Specification Restraints

The final step is to compile the results of our two simulations in order to formulate recommended hardware and software specifications. Most quantum key distribution systems implement a privacy amplification algorithm, so that even if Eve does recover a small amount of the exchanged information, it can be rendered useless. Usually, any recovery percent less than 11% can be rendered irrelevant by privacy amplification.[2] For this reason I wanted to choose specifications that would keep Eve's recoverable data percent below 11%.

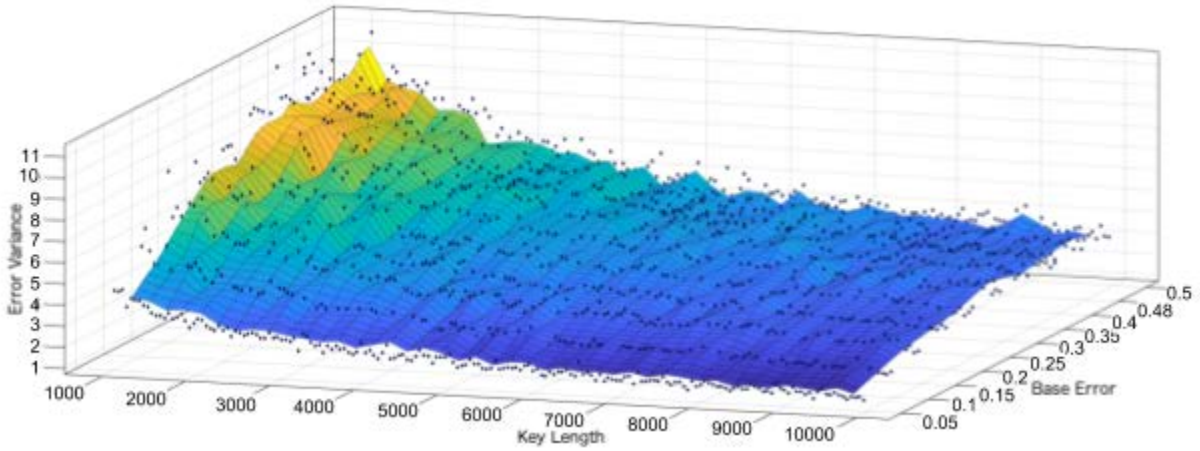

FIGURE 1. Visualization of how the key length and noise in cable affect the measured error variance

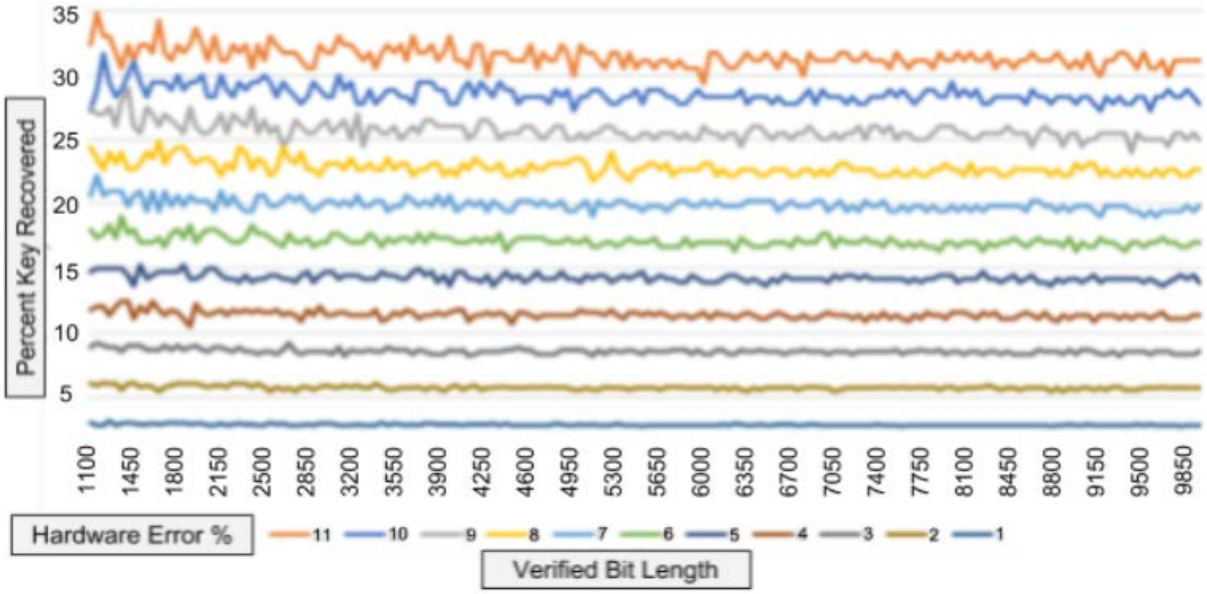

FIGURE 2. Visualization of the percent key recovered given measure rate and verified bit length

From our second simulation it can be seen that it is possible for Eve to recover just below 11% of the exchanged information with a detection rate of around 3.2%. This means that for privacy amplification to ensure any data Eve intercepts is useless, the error variance would ideally be less than 3.2%. Given x and y as the verified bit count and base hardware error, respectively, the equation must be less than 3.2%. For example, based on the graph shown, and a given hardware error rate of 10%, a 5000 bit verification would be sufficient while a 3000 bit verification would not. To make these results widely available I created an online calculator that would estimate the sufficiency of a user's proposed hardware error rate and verified bid count. This online calculator can be found at *http://chasekanipe.com/qkd.html.*

Error Bounds Calculations

One issue that arises from a distributed QKD system is the hardware noise variance. For example, if a large QKD network was implemented it is inevitable that the error bounds will vary from client to client due to the simple fact that error should initially increase linearly with distance. As I showed above, the protocol is most effective when the error bound is within around 3.2% of the actual error (because privacy amplification algorithms can reduce the usable bits to 0). For this reason, it would be necessary that the clients would have previously exchanged expected error information or highly reliable hardware.

CONCLUSION

Unless changes are made to the protocol, I conclude that it is impractical and insecure for distributed implementations. This insecurity is heightened by the fact that authentication difficulties leave QKD systems vulnerable to traditional MITM attacks. However, for dedicated use on high security data lines, it could be invaluable, especially with the incoming quantum computation threat to security and the necessity for long term data integrity. It could also be highly effective to encapsulate a classical scheme like RSA, or a post-quantum scheme so that even manipulation of the quantum channel could not immediately lead to data compromise.

REFERENCES

1. P. W. Shor and J. Preskill, *Physical Review Letters* **85**, 441 (2000).
2. B. Archana and S. Krithika, “Implementation of BB84 quantum key distribution using OptSim” *2nd Conference on Electronics and Communication Systems (ICECS)*, Coimbatore, 457-460 (2015).
3. N. H. Aizan, H. Zainuddin, and Z. A. Zukarnain, "Implementation of BB84 Protocol on 802.11i," *Network Applications, Protocols and Services, International Conference on Network Applications, Protocols, and Services (NetAPPS)*, Alor Setar, Kedah Malaysia, 130-134 (2010).
4. L. Susskind and A. Friedman, *Quantum Mechanics: The Theoretical Minimum*, Basic Books, 2014.