Optimal Cascade Block Size in BB84 Quantum Key Distribution

Supervisor: Prof. Hamzeh Roumani

Student: Abdulaziz Busbate

Department of Electrical Engineering & Computer Science, York University, Toronto, Canada

Introduction

Quantum mechanics makes it impossible (not just computationally infeasible) to passively eavesdrop on a communication channel; i.e. to listen in without being detected. Quantum channels are thus ideal for secret key distribution, and a protocol named BB84-Cascade has been devised to manage the transmission and correct channel errors.

Quantum Key Distribution

In Quantum key distribution, bits of information are encoded in polarized photons known as quantum bits (qubits). The security of QKD is in that an eavesdropper would be incapable of intercepting a qubit without altering it. Thus, the interception would be detectable by Alice and Bob. Moreover, quantum laws forbid an eavesdropper from cloning photons and performing multiple measurements.

BB84 Protocol

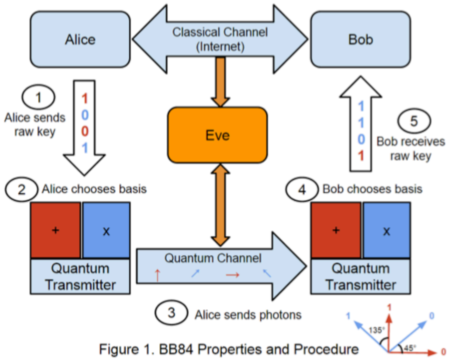


Figure 1. BB84 Properties and Procedure

In the BB84 protocol, two bases are used corresponding to four polarization states. The rectilinear basis (+) consists of 0˚ (⟶) and 90˚ (↑) polarizations, and the diagonal basis (x) consists of 45˚ (↗) and 135˚ (↖) polarizations. Moreover, measuring a qubit polarized in one basis will randomize the result if the other basis is used.

The first step in BB84 is called quantum transmission, where Alice generates a random bit string called the raw key and randomly chooses a polarization basis for each bit. Then, Alice uses the quantum channel to transmit one polarized photon for each bit of the raw key using its randomly chosen basis.

Bob receives the qubits and randomly chooses a basis to measure each photon. If Bob chooses the correct basis, he will obtain the same bit value that Alice sent. And if he chooses the wrong basis, he will still obtain the correct bit value 50% of the time.

If an eavesdropper intercepts a polarized photon and measured it in the wrong basis, she will randomize the state of the photon thus affecting Bob's measurement. Therefore, Bob will obtain the same correct bit value that Alice sent 25% of the time regardless of the basis he chooses.

The next step performed is a public discussion known as sifting. In sifting, Bob and Alice reveal their chosen bases on the classical channel. They then discard all bit positions for which Bob chose the wrong basis. This will produce Alice and Bob's sifted keys.

Next, Alice and Bob publicly share a random sample of their sifted keys to estimate the error rate (differences). Assuming channel noise is negligible, Alice and Bob's sifted keys should match except when an eavesdropper has intercepted the quantum channel. The estimated error rate will reveal Eve's presence in the quantum channel and estimate the percentage of qubits she intercepted.

Quantum Transmission

Alice's Photons:	↖	→	↖	↗	↗	↗	→	↖	↑	↑	↖	↗	↑	→	↖	↖	↗	→	↖	↑	⋯
Alice's Bases:	x	+	x	x	x	x	+	x	+	+	x	x	+	+	x	x	x	+	x	+	⋯
Alice's Raw Key:	1	0	1	0	0	0	0	1	1	1	1	0	1	0	1	1	0	0	1	1	⋯
Eve's Bases:	x	-	-	+	-	x	+	-	+	-	+	+	+	+	-	+	-	+	-	+	⋯
Eve's Raw Key:	1	-	-	0	-	0	0	-	1	-	1	1	1	0	-	1	-	0	-	1	⋯
Bob's Bases:	x	x	+	x	+	+	+	+	x	+	x	x	x	+	x	x	+	+	+	+	⋯
Bob's Raw Key:	1	1	1	1	1	0	0	0	1	1	1	1	1	0	1	1	1	0	0	1	⋯

Public Discussion (Sifting)

Sifting:	✓			✓			✓			✓	✓	✓		✓	✓	✓		✓		✓	⋯
Alice's Sifted Key:	1			0			0			1	1	0		0	1	1		0		1	⋯
Bob's Sifted Key:	1			1			0			1	1	1		0	1	1		0		1	⋯

Figure 2. BB84 Protocol

Error Reconciliation (Cascade Protocol)

The main purpose of error reconciliation is to correct errors in Bob's sifted key to reach a key identical to Alice's. A protocol named Cascade, performed entirely on the classical channel, is chosen to accomplish this task.

Cascade consists of several passes. In the first pass, Alice and Bob's sifted keys are divided into blocks of initial block size k. Then, each block's parity bit is calculated and compared. If parities agree, they assume no errors are present or an even number of errors are masked, so they move on to the next block. If parities disagree, they perform a binary search, diving the block into sub-blocks and so on, while comparing parities until an error is found and corrected.

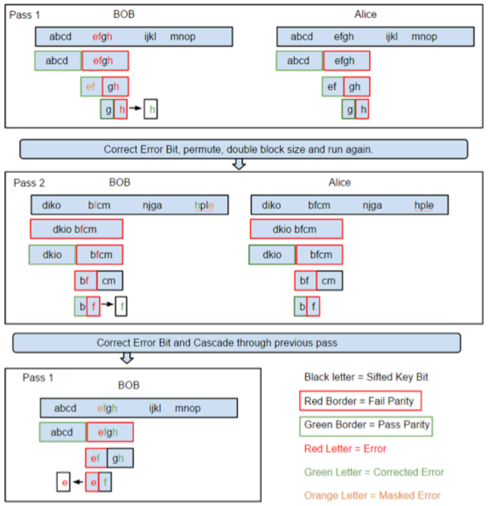


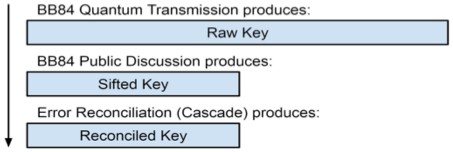
Figure 3. Binary Search and Cascade Operation

In the next pass, the block size k is doubled and the sifted keys are randomly permuted to distribute error locations uniformly. Then, the first pass is applied again but if an error is found in this pass, it will not only be corrected but also a Cascade operation is going to be performed.

If an error is found after the first pass, it means it was masked with another error(s) in pervious pass(es). The Cascade operation's job is to travel through previous pass(es) and binary search the corrected error's block to find the masked error. A Cascade operation will be performed for every error found whether the error is found by a binary search or another Cascade operation.

Passes will continue on until Alice and Bob are confident that no errors are left in Bob's sifted key.

Protocol Summary



BB84 Quantum Transmission produces:

Raw Key

BB84 Public Discussion produces:

Sifted Key

Error Reconciliation (Cascade) produces:

Reconciled Key

Figure 4. Protocol Summary

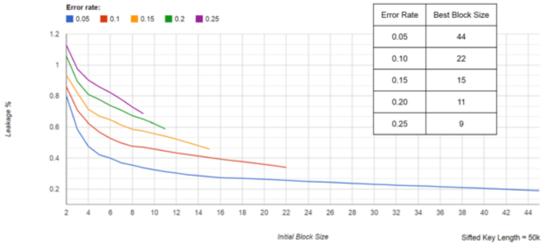
Information Leakage and Block Size Optimization

Determining the initial Cascade block size is critical due to its exponential effect on information leaked to an eavesdropper. The best initial block size k is when it's unlikely to have more than one error located in a single block. Using an initial block size smaller than k will exponentially increase the amount of leakage. On the other hand, using initial block sizes that are larger than k may prevent the passes from converging.

There are two types of information that gets leaked. When Eve measures a photon in the same basis as Alice, she learns one deterministic bit. And when the parity of a block is revealed, Eve learns one equational bit. Eve controls the deterministic bits based on how often she intercepts. Hence, we only need to minimize the equational bits leakage.

We simulated the entire protocol in a Java program and ran the simulation for a variety of error rates and initial block sizes. For each run, we tracked the number of bits leaked, both deterministically and equationally.

Figure 5 depicts the number of equational bits leaked versus the initial block size for five different error rates. As the initial block size increases, the equational bits leaked decreases. Also, as the initial block size increases, the number of passes needed increases. Since the block size doubles every pass, there is a maximum number of passes can be performed before the block size becomes as large as the sifted key. This explains why increasing the initial block size has to stop at some point depending on the sifted key's length.



Error Rate	Best Block Size
0.05	44
0.10	22
0.15	15
0.20	11
0.25	9

Figure 5. Equational Bits Leakage