An Accurate Analysis of the BINARY Information Reconciliation Protocol by Generating Functions

Sean Seet
DSO National Laboratories
Singapore S118230
Email: flamingarrow13@gmail.com

Ruth Ng Ii-Yung
University of Chicago
Chicago, IL 60637
Email: ruthfrancisng@uchicago.edu
(This research was done while the author was at DSO National Laboratories)

Khoongming Khoo
DSO National Laboratories
Singapore S118230
Email: kkhoongm@dso.org.sg

***Abstract*—Information Reconciliation (IR) protocols, which achieve error correction of shared secrets by public discussion, is an important process in Quantum Key Distribution (QKD). We provide an analysis of Brassard's BINARY and CASCADE IR protocols, two protocols commonly used in QKD. Using generating functions, we give an accurate result on BINARY. We derive the error probability distribution at each pass, which allows us to compute the decoding error probability and the number of "leaked" bits; two quantities crucial in the proof of security for QKD. We then corroborate the probability distribution computed by our formulas with actual simulation results. Finally we show that our formulas give better estimate for the decoding error probability of BINARY than the upper bound derived by Brassard for CASCADE. Because CASCADE should have better decoding performance than BINARY, this shows that Brassard's estimate of CASCADE may be too loose and can be improved. Our accurate formulas for BINARY can also be used as a basis on which to derive more accurate formulas for CASCADE.**

I. Introduction

Quantum Key Distribution (QKD) is an important technique to establish secret keys for secure communications. Its advantage over traditional cryptographic key distribution scheme is that QKD offers unconditional security while conventional schemes can only offer computational security.

In QKD, Alice and Bob first performs a photon exchange to share a secret string, called the raw key. However, there will be errors in the raw key they share because of:

1) Quantum channel noise.
2) Quantum eavesdropping by the adversary Eve.

Thus we need Information Reconciliation (IR), a 2-party error correction protocol, to correct the raw key through exchanging parity bits by public discussion. Because the adversary will learn some information of the raw key through quantum eavesdropping and observing the public exchange of parity bits, privacy amplification is used to compress the corrected raw key to a shorter final key to remove the adversary's knowledge.

The IR protocol used in the first practical instantiation of QKD is the BINARY IR protocol in [1]. It chops up a secret string into blocks, and perform binary search on each block to search out an error bit. The secret string is then permuted and this binary search error correction is performed over several passes until all error bits are corrected. Later, Brassard [3] introduced an improved version of BINARY, called CASCADE, which does backtracking to correct errors that was missed in binary search error correction of the earlier passes.

The CASCADE IR protocol is commonly studied and implemented because it corrects all errors by revealing the least number of parity bits (close to the theoretical limit), e.g. see [2], [3], [4], [5]. To analyze the security of a QKD system that uses the CASCADE IR protocol, we need to (e.g. see [6], [7]):

1) Measure the information leak of the raw key because of the exchange of parity bits.
2) Measure the decoding error probability, which is the probability that not all errors are corrected after completion of the IR protocol.

Both these values can be computed if we can predict the probability distribution of the number of error bits accurately. Till now the published results are either experimental [2], [4] or gives a loose estimate/bound [3].

Because the CASCADE protocol is built upon the BINARY IR protocol, see [3, Section 7.1]. One way to advance research in this direction is to first derive an accurate formula for the probability distribution of the BINARY IR protocol. We achieve this in Section II by an innovative application of the technique of generating functions. The accuracy of our formula is corroborated by matching it with simulation results of the BINARY protocol in Section III. From the probability distribution, we can derive both the information leak of the raw key and the decoding error probability.

In particular, we point out in Section IV that the decoding error probability of BINARY that we derived is already better (smaller) than the upper bound derived by Brassard for CASCADE in [3]. Keeping in mind that the CASCADE protocol is an improved version of BINARY, it should have a better (smaller) decoding error probability. Thus, the bound derived by Brassard may be too loose and can be improved. Our accurate formula for BINARY can be a basis on which to build further research to derive more accurate formula for CASCADE.

We define the following terms (to be used in the rest of this extended abstract) as follows:

- n is the length of the key.

- k_i is the block size at pass i.
- p is the probability that a specific bit will be incorrectly transmitted in the quantum channel.
- $\Delta_i(j-y \mid j)$ to be the probability that on the i^{th} pass, $j-y$ errors are corrected conditioned on the number of errors being j.
- $P_i(y)$ is the probability of there being y errors on the i^{th} pass.

II. An Accurate Analysis of BINARY

Theorem 1. *Let $P_i(y)$ be the probability that there are y errors left in the raw key after pass i of the BINARY IR protocol. Then the initial probability distribution before IR is given by:*

$$P_0(y) = \binom{n}{y} p^y (1-p)^{n-y}$$

and for $i \geq 1$, the probability distribution after correction in pass i is given by:

$$P_i(y) = \sum_{j=y}^{y+\frac{n}{k_i}} P_{i-1}(j)\Delta_i(j-y \mid j)$$

With this, we are left only to consider the calculation of $\Delta_i(j-y \mid j)$.

Theorem 2. *The quantity $\Delta_i(j-y \mid j)$, needed for the computation of Theorem 1, is given by:*

$$\Delta_i(j-y \mid j) = \frac{\binom{\frac{n}{k_i}}{j-y}}{\binom{n}{j}} \times C_{i,j,y}$$

Where $C_{i,j,y}$ = Coefficient of x^j in

$$\left(\frac{(1+x)^{k_i}-(1-x)^{k_i}}{2}\right)^{j-y}\left(\frac{(1+x)^{k_i}+(1-x)^{k_i}}{2}\right)^{\frac{n}{k_i}-(j-y)}$$

Notice that these equations are sufficient to find $P_i(0)$ given some i. Therefore, we can compute the decoding error probability $1-P_i(0)$, the chance that on the i^{th} pass not all errors have been corrected. From the probability distribution, we can also compute the number of leaked bits.

III. Comparison of our Computation with Simulation of BINARY

To corroborate our results, we ran 10000 trials on BINARY with $k_1 = 16$, $n = 256$, $p = 0.03$ and matched this against the probability distribution $P_i(j)$ computed from Theorems 1 and 2. The computation is done on the mathematical software MAPLE, while the simulation of BINARY is programmed in C. Due to space restrictions, here we display a truncated table of comparison in Table I.

We repeated the comparison for $k_1 = 16$, $n = 2048$, $p = 0.03$ in Table II.

From tables I and II, we see that our derivation of the probability distribution of BINARY by Theorems 1 and 2 is corroborated by actual simulation results.

We can also see that our theoretical expected value of the number of leaked bits is close to the average number of leaked bits in each pass from 10000 trials of BINARY simulation. This is shown in Tables III and IV.

	Pass $i=1$	Pass $i=2$	Pass $i=3$	Pass $i=4$
$P_i(0)$	0.25533	0.68058	0.86172	0.91689
$P_i(2)$	0.35897	0.23196	0.10989	0.06616
$P_i(4)$	0.24146	0.06757	0.02286	0.01372
$P_i(6)$	0.10347	0.01608	0.00457	0.00269
$P_i(8)$	0.03174	0.00319	0.00081	0.00047
$P_i(0)$	0.24960	0.68040	0.86090	0.91700
$P_i(2)$	0.36690	0.23110	0.10870	0.06530
$P_i(4)$	0.23750	0.06630	0.02430	0.01390
$P_i(6)$	0.10310	0.01910	0.00500	0.00310
$P_i(8)$	0.03440	0.00230	0.00080	0.00070

TABLE I
BINARY Calculation (top) and Simulation (bottom): $k_1 = 16$, $n = 256$, $p = 0.03$

	Pass $i=1$	Pass $i=2$	Pass $i=3$	Pass $i=4$
$P_i(0)$	0.00002	0.08808	0.63320	0.92559
$P_i(2)$	0.00020	0.17874	0.23883	0.06221
$P_i(4)$	0.00114	0.21261	0.08548	0.00974
$P_i(6)$	0.00421	0.19004	0.02891	0.00192
$P_i(8)$	0.01166	0.14029	0.00937	0.00042
$P_i(0)$	0.00000	0.08700	0.63740	0.92750
$P_i(2)$	0.00010	0.17690	0.23460	0.05950
$P_i(4)$	0.00110	0.21610	0.08540	0.01030
$P_i(6)$	0.00360	0.19370	0.02840	0.00230
$P_i(8)$	0.01070	0.13500	0.01020	0.00020

TABLE II
BINARY Simulation (top) and Calculation (bottom): $k_1 = 16$, $n = 2048$, $p = 0.03$

	Experimental	Theory
L_1	36.1064	36.1109
L_2	16.9910	16.9480
L_3	7.0888	7.0955
L_4	3.0052	2.9708

TABLE III
The expected amount of leaked information in the i^{th} pass, $k_1 = 16$, $n = 256$, $p = 0.03$

	Experimental	Theory
L_1	289.2384	288.8842
L_2	141.6640	141.3399
L_3	59.8124	59.8092
L_4	14.4960	14.5381

TABLE IV
The expected amount of leaked information in the i^{th} pass, $k_1 = 16$, $n = 2048$, $p = 0.03$

IV. Comparison of Decoding Error Probability with Brassard's Bound [3]

A. Brassard's Upper Bound on Decoding Error Probability

In Brassard's analysis [3] of CASCADE with an error probability of p, he focuses on the number of errors in block v of pass 1 (this block is called K_v^1 in [3]). After error correction in pass i, he would backtrack the error bits through to the first pass to investigate the number of errors remaining in the

original block K_v^1. He denotes the probability that there are $2j$ errors left in K_v^1 after pass i to be $\delta_i(j)$.

Proposition 1. *([3, Section 7.2]) In the* CASCADE *IR protocol with an error probability of p, the following inequality holds for* $\delta_i(j)$*:*

$$\delta_i(j) \leq \frac{\delta_{i-1}(j)}{2} \leq \frac{\delta_1(j)}{2^{i-1}} \text{ for } i, j > 0,$$

if the following two conditions hold:

$$k_1 p - \frac{(1-(1-2p)^{k_1})}{2} \leq -\frac{\ln(1/2)}{2},$$

and

$$\sum_{r=j+1}^{k_1/2} \delta_1(r) \leq \frac{1}{4}\delta_1(j) \text{ for all } j.$$

Using Proposition 1, we can compute the decoding error probability from $\delta_i(0)$ as follows.

$$P_i(0) = \delta_i(0)^{n/k_1} \geq \left(1 - \sum_{j=1}^{k_1/2} \frac{\delta_1(j)}{2^{i-1}}\right)^{n/k_1}.$$

Note that the above expression subtracted from 1 would give an upper bound for the decoding error probability, given by $1 - P_i(0)$.

B. Comparison to Brassard

In this section, we compute $P_i(0)$ in BINARY, based on Theorems 1 and 2, and compare it to Brassard's lower bound for $P_i(0)$ in CASCADE, based on Proposition 1 and the subsequent discussion. It can be verified that Brassard's bound can be applied because the two conditions of Proposition 1 are satisfied for $k1 = 16$ and $p = 0.03$. I.e.

$$16 \times 0.03 - \frac{(1-(1-2\times 0.03)^{16})}{2} \leq -\frac{\ln(1/2)}{2},$$

and

$$\sum_{r=j+1}^{8} \delta_1(r) \leq \frac{1}{4}\delta_1(j) \text{ for all } j.$$

The results are summarized in Tables V and VI:

Value	Our Calculation for BINARY	Brassard's Lower Bound for CASCADE
$P_1(0)$	0.25533	0.25533
$P_2(0)$	0.68058	0.51271
$P_3(0)$	0.86172	0.71854
$P_4(0)$	0.91688	0.84839

TABLE V
COMPARISON, $k_1 = 16$, $n = 256$, $p = 0.03$

We see that our calculation of $P_i(0)$ in BINARY is much larger than Brassard's lower bound for $P_i(0)$ in CASCADE. Equivalently, our calculation gives a much smaller decoding error probability $1 - P_i(0)$ for BINARY than the upper bound deduced from Brassard for CASCADE.

However, from the description of CASCADE in [3], it is obvious that CASCADE can correct many more errors for the same number of passes. Thus, CASCADE should have a much higher $P_i(0)$ or equivalently, a much smaller decoding error probability $1 - P_i(0)$, than BINARY.

Value	Our Calculation for BINARY	Brassard's Lower Bound for CASCADE
$P_1(0)$	0.00002	0.00002
$P_2(0)$	0.08808	0.00476
$P_3(0)$	0.63320	0.07106
$P_4(0)$	0.92559	0.26839
$P_5(0)$	0.98420	0.51894
$P_6(0)$	0.99492	0.72068
$P_7(0)$	0.99722	0.84902

TABLE VI
COMPARISON, $k_1 = 16$, $n = 2048$, $p = 0.03$

Thus, we conclude that Brassard's bound may be too loose. One possible reason for this discrepancy is that our analysis focuses on the global distribution of error bits, while Brassard's analysis focuses on the error distribution within a block. Thus when we extrapolate Brassard's result to all blocks, the bound will not be as tight. Thus, our methods here may yield a useful analysis of the global distribution of error bits in the CASCADE IR protocol.

ACKNOWLEDGMENTS

We would like to thank Chu-Wee Lim of DSO National Laboratories for suggesting the use of generating functions and provided C implementations of BINARY for us to verify our results.

REFERENCES

[1] Bennet, C.H.; Bessette, F.; Brassard, G.; Salvail, L., Smolin J. *Experimental Quantum Cryptography*. Journal of Cryptology, Vol. 5, No. 1 pp. 3-28, 1992.

[2] Bellot, Patrick; Dang, Minh-Dung *BB84 Implementation and Computer Reality. RIVF 2009* RIVF, pp. 1-8, 2009.

[3] Brassard, Gilles; Salvail, Louis *Secret-Key Reconciliation by Public Discussion. EUROCRYPT '93* Workshop on the theory and application of cryptographic techniques on Advances in cryptology, pp. 410-423, Springer-Verlag, 1994

[4] Calver, Timothy *An Empirical Analysis of the Cascade Secret Key Reconciliation Protocol for Quantum Key Distribution*, Thesis submitted to Airforce Institute of Technology, available from http://www.dtic.mil/dtic/tr/fulltext/u2/a549804.pdf.

[5] Capraro, I.; Occhipinti, T. *Implementation of a Real Time High Level Protocol Software for Quantum Key Distribution*, *ICSPC 2007*, IEEE International Conference on Signal Processing and Communications, pp. 704-707, IEEE Press, 2007.

[6] Renato Renner, *Security of Quantum Key Distribution*, PhD Thesis submitted to Swiss Federal Institute of Technology Zurich, available from http://arxiv.org/pdf/quant-ph/0512258v2.pdf.

[7] Valerio Scarani, Renato Renner *Security Bounds for Quantum Cryptography with Finite Resources*, Proceedings of TQC2008, Lecture Notes in Computer Science 5106 (Springer Verlag, Berlin), pp. 83-95 (2008).