Error Reconciliation in Quantum Key Distribution

Richard P. Brent
MSI, ANU

1 October 2009

Abstract

The problem of "error reconciliation" arises in Quantum Cryptography, which is more accurately described as Quantum Key Distribution (QKD). Alice and Bob each have copies of a secret key K, but Bob's copy is corrupted by errors. Alice and Bob want to reconcile their errors and agree on a secret key K' (possibly shorter than K), while disclosing as little information as possible to any eavesdropper Eve. In this introductory talk we outline the problem and describe a simple scheme that works well providing Bob's initial information about K is significantly better than Eve's.

The Scenario

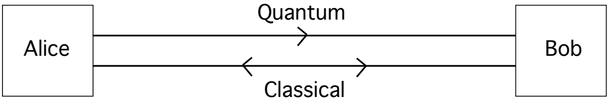
Alice and Bob communicate over two channels:

- A *quantum channel* (unidirectional from Alice to Bob)
- A *classical channel* (bidirectional)

The Scenario

Alice and Bob communicate over two channels:

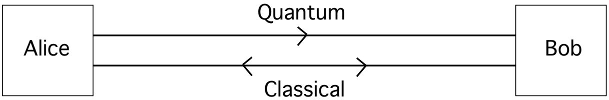
- A *quantum channel* (unidirectional from Alice to Bob)
- A *classical channel* (bidirectional)



The Scenario

Alice and Bob communicate over two channels:

- A *quantum channel* (unidirectional from Alice to Bob)
- A *classical channel* (bidirectional)



Eve can eavesdrop on both channels. However, if she eavesdrops on the quantum channel, this unavoidably changes some of the bits received by Bob. Alice and Bob can detect an eavesdropping attempt by monitoring the error rate on the quantum channel, and they can estimate how much information Eve could have gained.

The Assumptions

Eve has unlimited computational power (so she can crack conventional cryptography by brute force), but she has to obey the laws of physics!

The Objective

Alice and Bob want to share a secret key (a string of random bits) with the assurance that Eve has very little information about this key.

The Objective

Alice and Bob want to share a secret key (a string of random bits) with the assurance that Eve has very little information about this key.

They can use this secret key as a one-time pad for secure communication, or as a key for conventional cryptography.

The Objective

Alice and Bob want to share a secret key (a string of random bits) with the assurance that Eve has very little information about this key.

They can use this secret key as a one-time pad for secure communication, or as a key for conventional cryptography.

They may also keep some bits of the secret key for authentication purposes, so later messages can be signed. This avoids a “man in the middle” attack.

Key Expansion

Alice and Bob need to share some secret information at the start, so they can authenticate their messages on the classical channel and initialise their random number generators with the same seed.

Key Expansion

Alice and Bob need to share some secret information at the start, so they can authenticate their messages on the classical channel and initialise their random number generators with the same seed.

Thus Quantum Key Distribution (QKD) should really be called Quantum Key Expansion (QKE).

Key Expansion

Alice and Bob need to share some secret information at the start, so they can authenticate their messages on the classical channel and initialise their random number generators with the same seed.

Thus Quantum Key Distribution (QKD) should really be called Quantum Key Expansion (QKE).

Alice and Bob use quantum communication to turn a small secret into a larger one.

The Strategy – Error Reconciliation

Because the information received by Bob has some errors, these errors need to be corrected or discarded.

The Strategy – Error Reconciliation

Because the information received by Bob has some errors, these errors need to be corrected or discarded.

Note that the problem is not simply classical error correction, because

- Alice and Bob want to give away as little information as possible to Eve, who is assumed to be eavesdropping on the classical channel.

The Strategy – Error Reconciliation

Because the information received by Bob has some errors, these errors need to be corrected or discarded.

Note that the problem is not simply classical error correction, because

- Alice and Bob want to give away as little information as possible to Eve, who is assumed to be eavesdropping on the classical channel.
- Alice and Bob are free to discard some information rather than correct it, because all they need is *some* shared secret key, it does not matter exactly which key it is. Thus the process is called *error reconciliation* instead of *error correction*.

The Mathematical Model

Alice sends n random bits to Bob over a quantum channel. Each bit that Bob receives has a probability $p < 1/2$ of being incorrect.

The Mathematical Model

Alice sends n random bits to Bob over a quantum channel. Each bit that Bob receives has a probability $p < 1/2$ of being incorrect.

The errors could be due to noise and/or to the effect of eavesdropping by Eve. Initially Alice and Bob have an estimate of p. This estimate can be improved later, after they have some information to estimate the actual error rate.

The Physical Random Number Generator

It is important that Eve does not know the random number generator that Alice uses to generate her n random bits to send over the quantum channel – this random number generator should involve some random physical device so that it is unpredictable even if Eve has unlimited computational power.

The Pseudo-Random Number Generator

Alice and Bob share a pseudo-random number generator that is used to generate pseudo-random permutations. The seed for this random number generator could be part of their shared initial information, or could be sent during an earlier secure communication session.

The Pseudo-Random Number Generator

Alice and Bob share a pseudo-random number generator that is used to generate pseudo-random permutations. The seed for this random number generator could be part of their shared initial information, or could be sent during an earlier secure communication session.

If necessary, Alice could send Bob the key over the classical channel, *after* sending her random bits over the quantum channel. Although Eve is assumed to know the pseudo-random permutations, it is important that she can not predict them in advance, so she can not use them to decide which bits to intercept on the quantum channel.

Expected Distribution of Errors in Blocks

Alice and Bob choose a blocksize b depending on their common estimate of p. We assume $2 \le b \le n$.

Expected Distribution of Errors in Blocks

Alice and Bob choose a blocksize b depending on their common estimate of p. We assume $2 \leq b \leq n$.

For simplicity ignore the problem of what to do with the last block if b is not a divisor of n (since n is assumed to be large, whatever we do will make a negligible difference to the analysis).

The Random Permutation

Alice and Bob apply the same random permutation to their n-bit sequences, using their shared pseudo-random number generator. They should use a good random permutation algorithm.

The Random Permutation

Alice and Bob apply the same random permutation to their n-bit sequences, using their shared pseudo-random number generator. They should use a good random permutation algorithm.

Because of the first random permutation, we can assume that errors occurring in a block are independent, even if the original errors are correlated.

Analysis via Generating Functions

We use the generating function

$$G(x) = (px + q)^b,$$

where $q = 1 - p$. The coefficient of x^k in $G(x)$ gives the probability that a block of length b contains exactly k errors.

Analysis via Generating Functions

We use the generating function

$$G(x) = (px + q)^b,$$

where $q = 1 - p$. The coefficient of x^k in $G(x)$ gives the probability that a block of length b contains exactly k errors.

This probability is

$$p^k q^{b-k} \binom{b}{k},$$

but it is convenient to avoid expressions involving sums of binomial coefficients by working with $G(x)$.

Comparing Parity Bits

Alice and Bob compute the parities of their blocks, and compare parities using the classical channel. Thus, they can detect blocks with an odd number of errors.

Comparing Parity Bits

Alice and Bob compute the parities of their blocks, and compare parities using the classical channel. Thus, they can detect blocks with an odd number of errors.

We say that a block is *bad* if the computed parities disagree, and *good* if the parities agree. Note that a good block may contain an even number of errors.

Comparing Parity Bits

Alice and Bob compute the parities of their blocks, and compare parities using the classical channel. Thus, they can detect blocks with an odd number of errors.

We say that a block is *bad* if the computed parities disagree, and *good* if the parities agree. Note that a good block may contain an even number of errors.

Alice and Bob could use more sophisticated error detection/correction than simple parity bits, but it is not clear that this is desirable since it would disclose more information to Eve.

Discarding blocks/bits

- Alice and Bob discard their bad blocks.

Discarding blocks/bits

- Alice and Bob discard their bad blocks.
- They could use a binary search to locate and correct errors, but this is time-consuming and gives more information to Eve, so we adopt the simpler strategy, which works well if the block size is chosen correctly.

Discarding blocks/bits

- Alice and Bob discard their bad blocks.
- They could use a binary search to locate and correct errors, but this is time-consuming and gives more information to Eve, so we adopt the simpler strategy, which works well if the block size is chosen correctly.
- Alice and Bob also discard the first bit in each good block, in order to compensate for the parity information that Eve may have gained about the block.

Results from the Generating Function

- P_0 is the probability that a given block contains no errors:

$$P_0 = G(0) = q^b = (1 - p)^b \; .$$

Results from the Generating Function

- P_0 is the probability that a given block contains no errors:

$$P_0 = G(0) = q^b = (1-p)^b \,.$$

- P_1 is the probability that a block is bad (contains an odd number of errors):

$$P_1 = \frac{G(1) - G(-1)}{2} = \frac{1 - (1-2p)^b}{2} = bp + O(b^2 p^2) \,.$$

Results from the Generating Function

- P_0 is the probability that a given block contains no errors:

$$P_0 = G(0) = q^b = (1-p)^b \, .$$

- P_1 is the probability that a block is bad (contains an odd number of errors):

$$P_1 = \frac{G(1) - G(-1)}{2} = \frac{1 - (1-2p)^b}{2} = bp + O(b^2p^2) \, .$$

- P_2 is the probability that a block contains errors that are not detected (so it must contain an even number of errors):

$$P_2 = \frac{1 - 2(1-p)^b + (1-2p)^b}{2} = \frac{b(b-1)}{2}p^2 + O(b^3p^3) \, .$$

Number of Errors in a Good Block

The expected number of errors in a good block is

$$E_u = \frac{G'(1) - G'(-1)}{G(1) + G(-1)} ,$$

where the prime indicates differentiation with respect to x, so

$$G'(x) = bp(q + px)^{b-1} .$$

Thus

$$E_u = bp\left(\frac{1-(1-2p)^{b-1}}{1+(1-2p)^{b}}\right) = b(b-1)p^2 + O(b^3p^3) .$$

New Error Probability

After bad blocks have been discarded we expect the error probability for the remaining bits to be

$$\widetilde{p} = E_u/b = p\left(\frac{1-(1-2p)^{b-1}}{1+(1-2p)^b}\right) = (b-1)p^2 + O(b^2p^3)\ .$$

Rounds

The process of doing a permutation, comparing parities and discarding some bits is called a *round*. There will be several rounds, until Alice and Bob have agreed on a string of bits that is unlikely to contain any errors.

Rounds

The process of doing a permutation, comparing parities and discarding some bits is called a *round*. There will be several rounds, until Alice and Bob have agreed on a string of bits that is unlikely to contain any errors.

Actually, once Alice and Bob estimate that the expected number of errors remaining is $\ll 1$, they will, for reasons of efficiency, adopt a different strategy to confirm (or deny) that there are no remaining errors.

Re-estimation of Error Probability

Let E_b be the observed block error rate, that is the number of blocks in which an error is detected, normalised by the total number n/b of blocks. Thus the expectation $\mathcal{E}(E_b)$ of E_b is P_1, and we can obtain a new estimate p' of p.

$$E_b = \frac{1 - (1 - 2p')^b}{2}$$

(provided $E_b < 1/2$).

Re-estimation of Error Probability

Let E_b be the observed block error rate, that is the number of blocks in which an error is detected, normalised by the total number n/b of blocks. Thus the expectation $\mathcal{E}(E_b)$ of E_b is P_1, and we can obtain a new estimate p' of p.

$$E_b = \frac{1-(1-2p')^b}{2}$$

(provided $E_b < 1/2$). This gives

$$p' = \begin{cases} 0 & \text{if } E_b = 0, \\ \left(1-(1-2E_b))^{1/b}\right)/2 & \text{if } 0 < E_b < 1/2, \\ 1/2 & \text{otherwise.} \end{cases}$$

Re-estimation of Error Probability

Let E_b be the observed block error rate, that is the number of blocks in which an error is detected, normalised by the total number n/b of blocks. Thus the expectation $\mathcal{E}(E_b)$ of E_b is P_1, and we can obtain a new estimate p' of p.

$$E_b = \frac{1 - (1 - 2p')^b}{2}$$

(provided $E_b < 1/2$). This gives

$$p' = \begin{cases} 0 & \text{if } E_b = 0, \\ \left(1 - (1 - 2E_b)\right)^{1/b}\big) /2 & \text{if } 0 < E_b < 1/2, \\ 1/2 & \text{otherwise.} \end{cases}$$

This does not give an unbiased estimate of p, but it should be a consistent estimate and be close to correct if n/b is large.

The Blocksize

It is important to choose the blocksize not too large or small, to avoid inefficiency.

The Blocksize

It is important to choose the blocksize not too large or small, to avoid inefficiency.

We consider the case that there is little or no eavesdropping. The strategy discussed here may have to be modified if a substantial amount of eavesdropping is detected.

Effect of Discards

Discarding bad blocks reduces the number of bits from n to an expected $(1 - P_1)n$. Discarding one bit from each good block reduces this further, to $(1 - P_1)(1 - 1/b)n$.

Effect of Discards

Discarding bad blocks reduces the number of bits from n to an expected $(1 - P_1)n$. Discarding one bit from each good block reduces this further, to $(1 - P_1)(1 - 1/b)n$.

To partially compensate for this reduction, the “quality” of the bits should have improved. We can quantify this using Shannon’s coding theorem.

Estimating Bob's Information

The useful information (measured in bits) contained in Bob's initial n noisy bits is $(1 - H(p))n$, where

$$H(p) = -\left(p \log_2 p + q \log_2 q\right), \quad (q = 1 - p)$$

is the Shannon entropy, and p is the error probability.

Estimating Bob's Information

The useful information (measured in bits) contained in Bob's initial n noisy bits is $(1 - H(p))n$, where

$$H(p) = -\left(p \log_2 p + q \log_2 q\right), \quad (q = 1 - p)$$

is the Shannon entropy, and p is the error probability.
After discards the estimated error probability improves to $\widetilde{p}$, so Bob now has about

$$(1 - P_1)(1 - 1/b)(1 - H(\widetilde{p}))n$$

useful bits of information.

How to Choose the Blocksize

Dividing by n to normalize, define

$$J(b) = (1 - P_1)(1 - 1/b)(1 - H(\widetilde{p})) \ .$$

A reasonable criterion for choosing b is to maximise $J(b)$, subject to the constraints $2 \le b \le n$. The maximum occurs for $b \approx p^{-1/2}$ and can easily be obtained numerically.

How to Choose the Blocksize

Dividing by n to normalize, define

$$J(b) = (1 - P_1)(1 - 1/b)(1 - H(\widetilde{p}))\ .$$

A reasonable criterion for choosing b is to maximise $J(b)$, subject to the constraints $2 \le b \le n$. The maximum occurs for $b \approx p^{-1/2}$ and can easily be obtained numerically.

p	$p^{-1/2}$	b
0.2	2.24	2
0.1	3.16	3
0.05	4.47	5
0.01	10.0	10
0.001	31.6	32

Extreme case: *p* small

Recall that the expected error probability after the first round is

$$\widetilde{p} = p\left(\frac{1-(1-2p)^{b-1}}{1+(1-2p)^b}\right) .$$

It is interesting to consider two extreme cases. First, suppose that p is small and $b \approx p^{-1/2}$. Then

$$\widetilde{p} = p^{3/2} + O(p^2) .$$

This means that the error probability converges to zero rapidly (in fact superlinearly, with order $3/2$).

Extreme case: $p \approx 0.5$

Now consider the case that p is close to $1/2$, say $p = 1 - q = 1/2 - \varepsilon$, where ε is small but positive. In this case we can assume that $b = 2$. Write $\widetilde{p} = 1/2 - \widetilde{\varepsilon}$. We have

$$\widetilde{p} = \frac{p^2}{1 - 2p + 2p^2} = \frac{p^2}{p^2 + q^2} \,,$$

which gives

$$\widetilde{\varepsilon} = \frac{2\varepsilon}{1 + 4\varepsilon^2} \,.$$

Thus, when ε is small, $\widetilde{\varepsilon} \approx 2\varepsilon$. After about $\log_2(1/\varepsilon)$ rounds the error probability will no longer be close to $1/2$.

Number of Rounds

Combining the analysis of the extreme cases, we see that the probability that any errors remain is smaller than a given tolerance δ after about

$$\log_2\left(\frac{2}{1-2p}\right) + \log_{3/2}\log_2\left(\frac{n_f}{\delta}\right)$$

rounds, where n_f is the number of bits remaining after discards.

Predictions

The table gives the predicted behaviour if Alice and Bob start with $n = 10^6$ bits, and the error probability is $p = 0.25$. The errors are removed with five rounds, and at that point Alice and Bob share 99642 bits.

Predictions

The table gives the predicted behaviour if Alice and Bob start with $n = 10^6$ bits, and the error probability is $p = 0.25$. The errors are removed with five rounds, and at that point Alice and Bob share 99642 bits.

p	b	n	errors	bad blks	new n
0.250000	2	1000000	250000	187500	312500
0.100000	3	312500	31249	25416	157500
0.023810	7	157500	3749	3254	115470
0.003532	17	115470	407	385	102507
0.000201	71	102507	20	20	99642

Simulation Results

To confirm the predictions, we performed some simulations. The results of a typical run are given in the table. The simulation results are in good agreement with the predictions.

Simulation Results

To confirm the predictions, we performed some simulations. The results of a typical run are given in the table. The simulation results are in good agreement with the predictions.

p	b	n	errors	bad blks	new n
0.250000	2	1000000	250202	187552	312448
0.100100	3	312448	31325	25227	157844
0.023340	7	157844	3895	3409	114840
0.003921	16	114840	406	386	101872
0.000189	73	101872	20	20	99036

Predictions for Various p

The table shows the number of bits that we predict Alice and Bob should agree on, for an initial block of $n = 10^6$ bits and various error probabilities in the range $0.0001 \le p \le 0.49$.

p	final n
0.01	761620
0.10	318860
0.20	152151
0.30	56244
0.40	14880
0.45	3680
0.48	587

Conclusion

Mathematics, specifically generating functions, probability theory, and information theory, can be used to predict the performance of QKD and to choose optimal parameters.

References

Charles H. Bennett, François Bessette, Giles Brassard, Louis Salvail and John Smolin, Experimental quantum cryptography, *J. Cryptology* **5** (1992), 3–28.

Giles Brassard and Louis Salvail, Secret-key reconciliation by public discussion, *Advances in Cryptology – Eurocrypt 93*, *Lecture Notes in Computer Science* **765**, 1994, 411–423.

Jozef Gruska, *Quantum Computing*, McGraw-Hill, 1999. `http://www.fi.muni.cz/usr/gruska/quantum/`

Michael A. Nielsen and Isaac L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2000. `http://michaelnielsen.org/qcqi/`

Claude E. Shannon, *A Mathematical Theory of Communication*, University of Illinois Press, Urbaba, Illinois, 1949 (reprinted 1998). See also `http://en.wikipedia.org/wiki/Shannon_limit`.

Vikram Sharma, *Informatic Techniques for Continuous Variable Quantum Key Distribution*, PhD thesis, ANU, October 2007. (Includes a good bibliography on quantum cryptography.)

Richard Taylor, Near optimal unconditionally secure authentication, *Proc. Eurocrypt 1994*, LNCS **950**, Springer-Verlag, 1995, 244–253.

M. N. Wegman and J. L. Carter, New hash functions and their use in authentication and set equality, *J. Computer and System Sciences* **22** (1981), 265–279.