# Simulation of an entanglement-based quantum key distribution protocol: supplemental document

June 28, 2024

**Abstract**

In order to describe in detail the methods for this work we will first introduce here some primitives defined in [1] and then proceed to describe the two IR protocols that we used.

## BINARY

The `BINARY` primitive is an error-correction method based on a divide-and-conquer approach: given a couple of strings with mismatching parities, this string are divided in halves, whose parities are computed and compared to find in which half the error is. Then the procedure is repeated recursively until the length of the sub-strings with mismatching parity is 1 and then the error is identified and corrected by flipping the bit in one of the two strings (say Bob's). Please note that this primitive is interactive and it requires the public exchange of a certain number of parity bits equal to $\lceil \log n \rceil$ if $n$ is the length of the original string.

## CONFIRM

When applied to a couple of strings, this primitive will always return a positive outcome if the strings are identical. Instead, if there is disagreement between the two strings, it will report the presence of an error with only 50% probability. It consists in 1) extracting a random subset of bits (each bit is included in the subset with 50% probability), 2) comparing the parities and 3) raising a flag if there is a mismatch. This implies that if `CONFIRM` does not detect a mismatch between the parities of the sub-blocks for $k$ consecutive times, there is a $1 - 2^{-k}$ probability that the two strings coincide.

## BICONF

The $\texttt{BICONF}^s$ primitive has been defined in [2] but the original Cascade protocol there proposed does not make use of it. It consists in $s$ consecutive runs of `CONFIRM`, combined with the execution of a `BINARY` every time that an error is detected.

## BBBSS

Based on the QBER estimated in the IR step, the sifted key is divided in blocks containing $k = a/QBER$ bits (no recommendation regarding the value of $a$ was given in the original paper [1], but the choice of $a \approx 0.73$ for the similar Cascade protocol was later found to be optimal [2]). After that, `BINARY` is executed on each block, always discarding the last bit of any sub-block whose parity is publicly declared, so as to give up no information to Eve in this step. The procedure is iterated on the remaining key, dividing the key in blocks twice as long as in the first iteration. After that, as a further guarantee that the reconciliated keys are identical, `BICONF`[1] can be iterated (again sacrificing a bit after every pass) until it highlights no parity mismatches for $s$ consecutive times: this provides a $1 - 2^{-s}$ confidence that no errors survived the protocol. For this work we took $s = 10$.

## Cascade

The first iteration (or pass) of this protocol [2] is identical to the first iteration on BBBSS, except for the fact that – as in the rest of this protocol – no bits are discarded. Eve will then get some information about sub-block parities but this allows for the correction of more than one error per parity exchange. For any $i$-th pass after the first one, the block size $k_i$ is doubled, and whenever an error is found the following routine is executed: after that Bob receives Alice's parities and finds a disagreement when comparing them with its own, he executes `BINARY` to correct the mismatching bit $B_l$. Then he builds a list $\mathcal{K}$ of all the blocks that contained $B_l$ in the previous iterations and performs `BINARY` on the shortest of such blocks, correcting another bit $B_{l'}$. At this point $\mathcal{K}$ is updated:

$$\mathcal{K} \to \mathcal{K}' = (\mathcal{B} \cup \mathcal{K}) \setminus (\mathcal{B} \cap \mathcal{K}) \tag{1}$$

$\mathcal{B}$ being the list of blocks containing $B_{l'}$. This corresponds to adding to $\mathcal{K}$ the blocks containing $B_{l'}$ while subtracting the ones containing both $B_l$ and $B_{l'}$. The previous steps are repeated until $\mathcal{K} = \emptyset$. If any other mismatch is found between Alice's and Bob's parities of the $i$-th iteration these steps are repeated. Finally another iteration can be performed if considered necessary.

# References


[1] C. H. Bennett, F. Bessette, G. Brassard, L. Salvail, and J. Smolin, "Experimental quantum cryptography," Journal of Cryptology p. 26 (1992).

[2] G. Brassard and L. Salvail, "Secret-Key Reconciliation by Public Discussion," in *Advances in Cryptology — EUROCRYPT '93,* vol. 765 T. Helleseth, ed. (Springer Berlin Heidelberg, Berlin, Heidelberg, 1994), pp. 410–423.