Towards an Optimal Implementation of Cascade

J. Martinez-Mateo[1], C. Pacher[2], M. Peev[2], A. Ciurana[1], V. Martin[1]*

[1] Universidad Politécnica de Madrid
Campus de Montegancedo, 28660, Boadilla del Monte (Madrid), Spain
[2] Safety & Security Department, AIT Austrian Institute of Technology GmbH
Donau-City-Strasse 1, 1220 Vienna, Austria.
*e-mail: vicente@fi.upm.es   web: http://gcl.ls.fi.upm.es

Abstract.— Cascade is an information reconciliation protocol proposed in the context of secret key agreement in quantum cryptography. This protocol allows removing discrepancies in two partially correlated sequences that belong to distant parties, connected through a public noiseless channel. It is highly interactive, thus requiring a large number of channel communications between the parties to proceed and, although its efficiency is not optimal, it has become the de-facto standard for practical implementations of information reconciliation in quantum key distribution. The aim of this work is to analyze the performance of Cascade, to discuss its strengths, weaknesses and optimization possibilities, comparing with some of the modified versions that have been proposed in the literature. When looking at all design trade-offs, a new view emerges that allows to put forward a number of guidelines and propose near optimal parameters for the practical implementation of Cascade improving performance significantly in comparison with all previous proposals.

Table 1. Parameters used for the original, modified and optimized versions of Cascade. A frame length of $n = 10^4$ bits was considered except for the optimization labeled as (8) where the frame length is $n = 2^{14}$.

Protocol	Block sizes (approx.)			Cascade	BICONF	Block
	k_1	k_2	k_i	passes		reuse
orig. [1]	$0.73/Q$	$2k_1$	$2k_{i-1}$	4	no	no
mod. (1) [2]	$0.92/Q$	$3k_1$	–	2	yes	no
opt. (3)	$1/Q$	$2k_1$	$n/2$	16	no	no
opt. (4)	$1/Q$	$2k_1$	$n/2$	16	no	yes
...	*results not included here*					
opt. (7)	$2^{\lceil \log_2 1/Q \rceil}$	$4k_1$	$n/2$	14	no	yes
opt. (8)	$2^{\lceil \alpha \rceil}$	$2^{\lceil (\alpha+12)/2 \rceil}$	$n/2$[a]	14	no	yes

[a] $\alpha = \log_2(1/Q) - \frac{1}{2}$, $k_3 = 2^{12} = 4096$ and $k_i = n/2$ for $i > 3$.

Initially, the original Cascade [1] is compared to the modified protocol described in [2], that uses two passes of Cascade and subsequent iterations of BICONF. Efficiency, channel uses and frame error rate (Figs. 1, 2 and 3, respectively) have been exhaustively computed for a base frame length of $n = 10^4$ bits (as in [1, 2], which allows for a fair comparison).

Fig. 1 shows that the efficiency of the modified version of Cascade improves for this frame length when $Q > 0.5\%$. Other frame lengths have also been computed. For these, Cascade's efficiency does not improve while it does, although marginally, for the modified version.

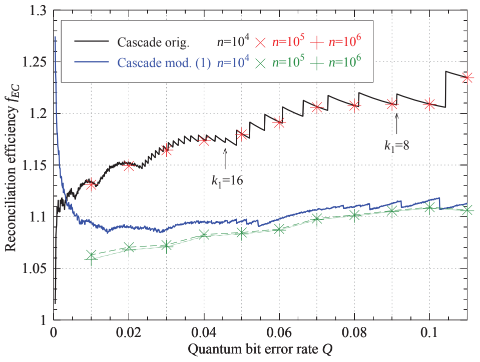

Fig. 1. Reconciliation efficiency, $f_{EC} = (1-R)/h(Q)$, for the original Cascade [1] and the modified version in [2] labeled as (1).

Efficiency and channel uses curves for both protocols exhibit a sawtooth behavior due to the discreteness of the block sizes. Jumps occur at those values of Q where k_1 changes its integer value, and subsequently k_2, etc. Some of these are marked in Figs. 1 & 2. The effect of k_2 is clearly seen in Fig. 2 for the modified version of Cascade as a smaller amplitude sawtooth behavior seen for the same value of k_1.

Fig. 2 also shows that the price to pay for improving the efficiency is an increase (a significant one) in the number of channel uses. By channel uses we mean the number of communication rounds or pair of messages exchanged through the noiseless channel to disclose parity values.

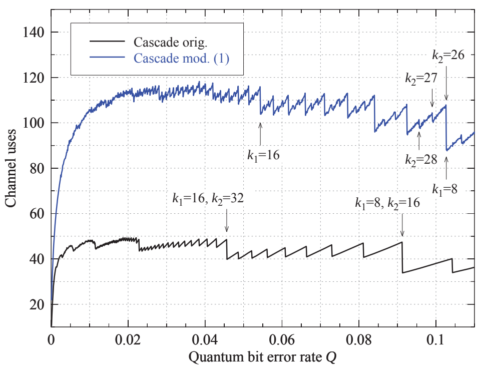

Fig. 2. Number of channels uses or communication rounds.

However, in the next figure we show that there also exist a discrepancy in the failure probability of the reconciliation protocol, or frame error rate (FER). Fig. 3 shows now that the FER is significantly higher for the modified version of Cascade. Therefore, although the efficiency improves, the fraction of successfully reconciled frames worsens. Different frame lengths have been considered and compared, and it is evident that while the FER with the frame length in Cascade, this is not the case for the modified version, for which for lengths of 10^5 bits the FER remains remarkably constant at 10^{-3}.

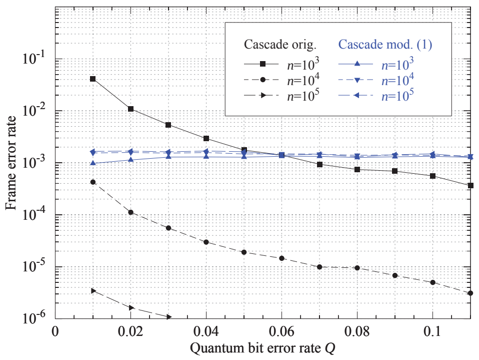

Fig. 3. Frame error rate (= failure probability).

Next, we studied the ability of Cascade to adapt to variations in the communication channel. Simulations have been carried out using two different input parameters: (i) the error rate p used to initialize the protocol, i.e., the first block size k_1 is now derived from p and not from Q; and (ii) Q the actual quantum bit error rate. Note that p may stand for a (poor) estimate of Q. Therefore, Fig. 4 shows how the protocol behaves under time-varying channel conditions. These give more insight about some parameters used in the protocol (e.g., block sizes) and suggests possible optimizations.

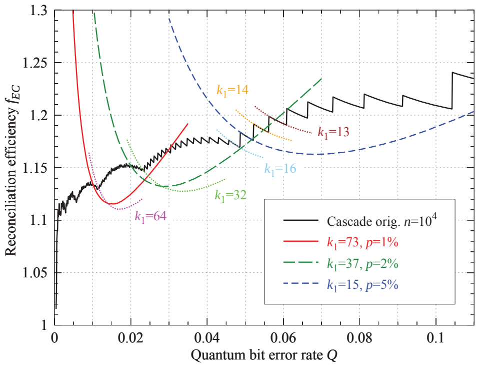

Fig. 4. Reconciliation efficiency, $f_{EC} = (1-R)/h(Q)$, of Cascade with constant first block size k_1.

As shown, the efficiency improves for $Q > p$. A price to pay for a better efficiency is a sharp increase in the number of exchanged messages (not shown), due to more errors being detected and corrected during the later algorithm passes. However, the FER is not significantly affected. Thus, we empirically show that the efficiency of the original Cascade is optimal for the three cases $p = 1\%$, 2% and 5% when $Q \approx 1.46\%$, 2.85% and 6.87%, respectively. Taking into account that the FER does not significantly increase, and disregarding the channel uses, it follows that the block size $k_1 \approx 1/Q$ is presumably optimal.

Then, a modification of [2] is proposed by replacing BICONF for a number of passes of Cascade with block size half of the frame length, as already hinted in [3], but using the first block size suggested as optimal in our previous simulations. This optimized version of Cascade is labeled as (3) in Fig. 6 and Table 1. Further, we also consider the subblock reuse proposed in [3], optimized version labeled as (4).

We also use a description of the information leakage that takes into account the frame error rate ε_{EC}, $\text{leak}_{EC} = (1-\varepsilon_{EC})(1-R) + \varepsilon_{EC}$, to optimize the number of passes. Fig. 5 shows how the leakage improves with the number of passes. A description of the efficiency that considers this leakage can be also used to look for the optimal block sizes, thus penalizing those parameters with high FER values.

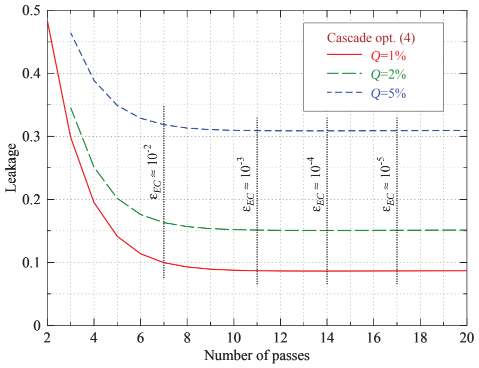

Fig. 5. Information leakage as a function of the number of passes.

Finally, to find the optimal block sizes that minimize the efficiency we use a Compass search algorithm. This showed that the optimal efficiency is obtained in most cases for k_1 and k_2 values that are powers of two or nearby values. Results with the optimized block sizes are shown in Fig. 6, version labeled as (7), and the values for the block sizes are given in Table 1. Consequently, an optimization was later computed for a power of two frame length $n = 2^{14}$ and considering also the third block size k_3. The results for this frame length shows even more convincingly, the importance of using power of two block sizes. In fact, this is even more important than any other protocol optimizations to improve the average reconciliation efficiency of Cascade. Results with the suggested parameters and a frame length of 2^{14} are also shown labeled as (8), and the block sizes are also given in Table 1. For a further description see [4].

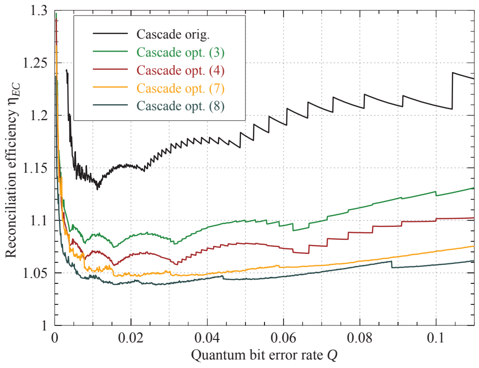

Fig. 6. Reconciliation efficiency, $\eta_{EC} = ((1-\varepsilon_{EC})(1-R) + \varepsilon_{EC})/h(Q)$, for the optimized versions of Cascade proposed in [4].

Conclusions.—

We provide a comprehensive comparison of the Cascade reconciliation protocol and some of its modified versions that have been proposed in literature. Results of exhaustive simulation have been used to compare the efficiency, communication rounds and frame error rate for all discussed versions. Based on the analysis of our results, we also propose an optimized version of Cascade that utilizes previous ideas, and leads to a near optimal implementation of the protocol.

Acknowledgments.— Partially supported by the projects Hybrid Quantum Networks, TEC2012-35673, funded by *Ministerio de Economía y Competitividad*, Spain, and HiPANQ, ICT10-067, Vienna Science and Technology Fund (WWTF).

References.

[1] G. Brassard and L. Salvail, "Secret-key reconciliation by public discussion," in *EUROCRYPT '93*, vol. 765, 1994, pp. 410–423.

[2] T. Sugimoto and K. Yamazaki, "A study on secret key reconciliation protocol "Cascade"," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E83-A, no. 10, pp. 1987–1991, 2000.

[3] H. Yan *et al.*, "Information reconciliation protocol in quantum key distribution system," in *ICNC 2008*, vol. 3, 2008, pp. 637–641.

[4] J. Martinez-Mateo *et al.*, "Demystifying the information reconciliation protocol Cascade," *arXiv:1407.3257 [quant-ph]*, 2014.