New Approaches to Increase Efficiency of Cascade Information Reconciliation Protocol

Metin Toyran
Electronics and Communication
Department, Engineering Faculty
Kocaeli University
İzmit, Kocaeli, Turkey
Email: metintoyran@gmail.com

Mustafa Toyran
National Research Institute
of Electronics and Cryptology
TÜBİTAK - BİLGEM
Gebze, Kocaeli, Turkey
Email: mustafa.toyran@tubitak.gov.tr

Sıtkı Öztürk
Electronics and Communication
Department, Engineering Faculty
Kocaeli University
İzmit, Kocaeli, Turkey
Email: sozturk@kocaeli.edu.tr

***Abstract*—In this abstract, we present ways to improve efficiency performance of Cascade error reconciliation protocol. Our ideas are based on using i) known bits and ii) known parities obtained during the execution of the protocol. We use this information to get rid of parity checks and run error corrections on smaller blocks. Computer simulations show that Cascade with these improvements is currently more efficient than both all the previous Cascade versions and other non-Cascade methods proposed for quantum key distribution information reconciliation.**

***Keywords:* Discrete-variable quantum key distribution (DV-QKD), secret key reconciliation (SKR), information reconciliation (IR), post-processing, Cascade**

I. Introduction

Cascade protocol is an information reconciliation (IR) method proposed firstly for use in quantum key distribution (QKD) in 1993 [1]. For an IR method in QKD, one of the main performance measures is efficiency which depends on the amount of exchanged information to make reconciliation possible. Since this redundant information is about keys that must be kept secret from unintended parties and transmitted over public eavesdroppable channels, it can damage the secrecy of keys. Therefore, more efficient, that is revealing less information, IR methods are needed for QKD.

Since its born, three noticable work increasing the efficiency performance of Cascade more and more are published in [2]–[4], statistically one per almost every seven/eight years. Currently, the most efficient version is the one published in [5]; however, this protocol is not Cascade anymore since it has major changes.

In this work, we implement the strategy given in [3] which is still Cascade and apply our improvements. Simulation results show that the resulting Cascade has the highest efficiencies as far as we have seen.

II. Our Improvements

To decrease exchanged information and increase efficiency performance, we should aim at searching for errors in blocks as small as possible as performed in [3]. However, there are still other ways to make the blocks smaller than accomplished in that work; such as, using exactly known bits and known parities.

A. Exactly Known Bits

As illustrated in Figure 1 below, knowing the parity of a size-two block, and the value of the corrected bit, tells us the value of the other bit as well.

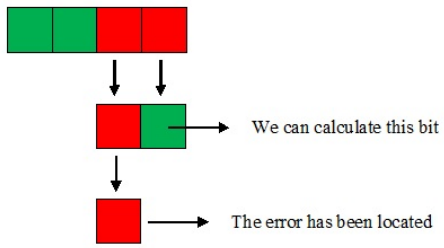

Fig. 1. Size-two blocks: Both two bits are exactly known.

Red ones are the parity mismatching blocks that contain odd number of errors (1, 3, 5, etc.) and green ones are the parity matching blocks that contain even number of errors (0, 2, 4, etc.) in the figures.

As illustrated in Figure 2, we can also get the values of all the three bits in a size-three block. We always take the bigger half as left branch in our implementations. And, as illustrated in Figure 3, only one bit will be exactly known in a size-three block when the error is located in the right branch (in our implementations).

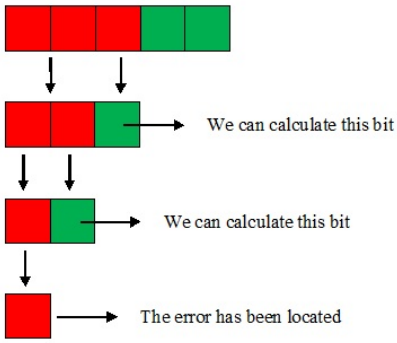

Fig. 2. Size-three blocks: Left branching case. All the three bits are exactly known.

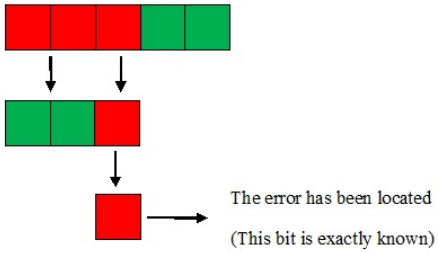

Fig. 3. Size-three blocks: Right branching case. Just one bit is exactly known.

B. Known Parities

During the protocol, many blocks are created. Before starting error correction on a block, the (distant) parties can remove the previously parity calculated smaller blocks where they agree on and that are included in the error detected block, as shown in Figure 4.

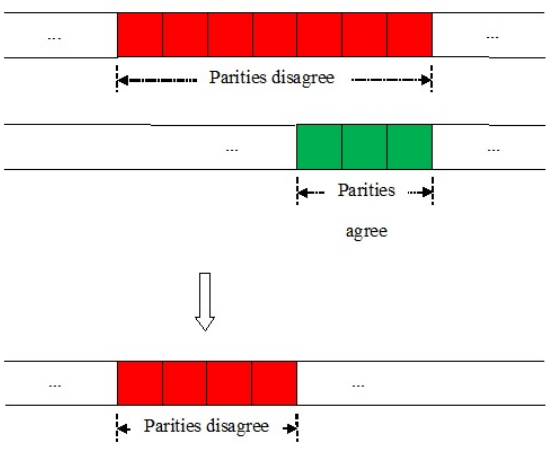

Fig. 4. Remove included smaller blocks where the parties agreed on the parity previously. This makes the error detected block smaller.

Applying all these improvements will make Cascade get rid of some of the parity checks and make some of the error detected blocks smaller just before applying error correction, named BINARY, on them.

C. New Parity Check

Based on the previous ideas, our new error detection strategy on the blocks can be summerized as follows:

- If the block consists of exactly known bits only or its whole parity can be calculated using the information in sections II-A and II-B, then no parity check is done.
- Otherwise, it runs parity check on the block. If parity mismatches, our new Binary method is executed on the block.
- Otherwise,
  - If block size is just 1 bit, this bit is also exactly known and it is recorded.
  - If all other bits of the block are exactly known, the remaining 1 bit is also exactly known (using the block parity) and it is recorded.

D. New BINARY

Based on the previous ideas, our new error correction strategy on the error detected blocks is given as follows:

- First, all the included parity matching blocks and exactly known bits recorded previously are removed from the block.
- Next, BINARY error correction is run on the remaining smaller block.
- Index of the corrected bit is returned.

E. Efficiency

Let A and B represent the initial secret key bit strings of length N at the sender's and the receiver's sides respectively, and p is the error rate of the (public) quantum channel. Then the conditional Shannon entropy between the two (partially) correlated random variables A and B is given as,

$$H(A|B) = Nh(p) \tag{1}$$

where $h(p)$ is binary entropy function. At least $Nh(p)$ redundant bits must be exchanged between the sender and the receiver for successful agreement of their key sequences.

If we define E as the average number of exchanged parities, then one measure of the efficiency is defined as follows,

$$\mu = 1 - \frac{E}{N} \tag{2}$$

TABLE I
COMPARISON OF OUR MORE EFFICIENT CASCADE IMPLEMENTATIONS WITH [4]

p (%)	k_1	k_2	k_3	β	Our β	η	Our η	FER	Our FER	η_{FER}	Our η_{FER}
1	128	512	4096	0.9963	**0.9974**	1.04219	**1.02944**	$8x10^{-5}$	$2x10^{-4}$	1.0431	**1.03171**
2	64	512	4096	0.9934	**0.9948**	1.04006	**1.03146**	$9.3x10^{-5}$	0	1.04062	**1.03146**
3	32	512	4096	0.9906	**0.9926**	1.03902	**1.03050**	$1.1x10^{-4}$	$4x10^{-4}$	1.03945	**1.03214**
4	32	256	4096	0.9862	**0.9895**	1.04313	**1.03255**	$9.4x10^{-5}$	$1x10^{-4}$	1.04342	**1.03286**
5	16	256	4096	0.9827	**0.9875**	1.04313	**1.03090**	$8.9x10^{-5}$	$2x10^{-4}$	1.04335	**1.03140**
6	16	256	4096	0.9777	**0.9843**	1.0458	**1.03221**	$1.1x10^{-4}$	$2x10^{-4}$	1.04601	**1.03262**
7	16	256	4096	0.9709	**0.9802**	1.0505	**1.03428**	$8.7x10^{-5}$	$2x10^{-4}$	1.05065	**1.03462**
8	8	256	4096	0.9632	**0.9751**	1.05465	**1.03691**	$9.7x10^{-5}$	$2x10^{-4}$	1.05479	**1.03720**
9	8	256	4096	0.9575	**0.9730**	1.05486	**1.03481**	$1.0x10^{-4}$	$1x10^{-4}$	1.05499	**1.03494**
10	8	256	4096	0.9493	**0.9696**	1.05736	**1.03441**	$1.0x10^{-4}$	$7x10^{-4}$	1.05747	**1.03518**
11	8	256	4096	0.9387	**0.9647**	1.0613	**1.03527**	$1.0x10^{-4}$	$4x10^{-4}$	1.06139	**1.03566**

A second measure of the efficiency used in the literature is calculated based on the capacity of the communication channel (Shannon limit) and given as,

$$\beta = \frac{\mu}{1 - h(p)} \tag{3}$$

And, a third measure of efficiency based on the $Nh(p)$ limit can is defined as,

$$\eta = \frac{E}{Nh(p)} = \frac{1 - \mu}{h(p)} \tag{4}$$

where η indicates the percentage of additional information revealed over the limit.

To analyze the robustness of secret key reconciliation (SKR) methods, frame(FER) and bit error rates(BER) are used. It is remarkable that higher efficiency values may not be significant in higher FER values due to discarding high number of corrupted frames. Therefore, in the presence of FER, η can be calculated as follows:

$$\eta_{FER} = \frac{(1 - FER)(1 - \mu) + FER}{h(p)} \tag{5}$$

In equation 5, the $(1 - FER)$ multiplier represents successfully reconciled frames, and $(1 - \mu)$ is the ratio of information revealed in reconciliation of erroneous bits.

III. EXPERIMENTAL RESULTS

We ran our Cascade implementation with the new parameter set given in [4], that is, $Rounds$: 14, k_1, k_2, k_3: as in Table I, $k_i : N/2, \quad 4 \leq i \leq Rounds$ for 10^4 frames of length 2^{14}.

In [4], the authors mentioned that their modified Cascade version, with their improvements and optimal parameter set, had the best efficiency values in the literature up to now. As seen from the Table I, our Cascade version is more efficient than that work.

IV. CONCLUSION

In this abstract, we presented several new ideas to increase the efficiency performance of Cascade. According to the results, our implementation of Cascade with the improvements mentioned above is more efficient than all these LDPC [6], Polar codes [7] and Cascade based approaches.

V. FUTURE WORKS

In this work, we apply our improvements on the smallest error detected block. Applying the improvements on all the error detected blocks and running BINARY on the smallest of the resulting blocks can also be tried. Also, in known parities improvement case, we remove the fully included parity matching blocks. Removing also partly included parity matching blocks can be tried, too.

REFERENCES

[1] Brassard G, Salvail L. (1993), *Secret key reconciliation by public discussion*, EUROCRYPT, Lofthus, Norway. New York, NJ, USA, Springer. pp. 41023.

[2] Sugimoto T, Yamazaki K. (2000), *A study on secret key reconciliation protocol Cascade*, Ieice T Fund Electr, E83A: 19871991.

[3] Yan H, Ren T, Peng X, Liu T, Guo H. (2008), *Information reconciliation protocol in quantum key distribution system*, IEEE Fourth International Conference on Natural Computation, Jinan, China, New York, NY, USA, 3: 637641.

[4] Mateo JM, Pacher C, Peev M, Ciurana A, Martin V. (2015), *Demystifying the Information Reconciliation Protocol Cascade*, QIC, Vol. 15, No:5&6 0453-0477.

[5] Pacher C, Grabenweger P, Mateo JM, Martin V. (2015), *An Information Reconciliation Protocol for Secret-Key Agreement with Small Leakage*, ISIT, 730-734.

[6] Elkouss D, Leverrier A, Allaume R, Boutros JJ. (2009), *Efficient reconciliation protocol for discrete-variable quantum key distribution*, IEEE International Symposium on Information Theory, Seoul, Korea. New York, NY, USA. pp. 18791883.

[7] Jouguet P, Kunz-Jacques S. (2014), *High performance error correction for quantum key distribution using polar codes*, Quantum Inf Comput, 14: 329338.