An Information Reconciliation Protocol for Secret-Key Agreement with Small Leakage

Christoph Pacher¹, Philipp Grabenweger¹, Jesús Martínez-Mateo², Vicente Martín²

¹ Digital Security & Safety Department, AIT Austrian Institute of Technology GmbH, 1220 Vienna, Austria
² Madrid Center for Computational Simulation, Universidad de Madrid, Spain

SUMMARY

We report on our information-theory based tuning approach (partly discussed in [3]) of the so-called Cascade [1] protocol to achieve very small leakage: We prove that powers of two are optimal values for the number of transmitted bits in the initial blocks (Box 3). This confirms and explains results of recent numerical optimizations [2]. Bits are corrected separately, meaning that their individual error probability in the second pass (round); corrected bits are fully taken into account (Box 4). Simulation results for efficiency and throughput of these optimizations are shown in Box 5 & 6. A significant improvement for the efficiency is obtained, although at a highly increased number of exchanged messages. Also in Box 5 & 6 variants with still very high efficiency but also high throughput are shown.

The leakage is for block sizes of 2^16 typically only 2.5% above the Shannon limit, and notably, this holds for an error rate p between 1% and 50%. For p between 1% and 6% the leakage is only 2% above the Shannon limit. As comparison, the leakage of the original Cascade algorithm is 20% (40%) above the Shannon limit for p of 10% (35%).

4 CATEGORISATION AND FURTHER IMPROVEMENTS EFFICIENCIES

Dichotomic Searches

- Recursively dividing only blocks with odd parity we localize one faulty bit.
- What about blocks with same parity? Not used so far.

Analysis of Blocks with Same Parity

- Starting with k = 2^k, approximately in 50% of the cases Alice’s and Bob’s blocks have the same parity.
- During the dichotomic search in each 2^k block with different parity we learn k - 1 blocks that have same parity (with sizes 2^k - 1, 2^k - 2, ..., 2).
- Conditional bit error probability in a block of size k with same parity

\[ \delta(k, p) \approx \frac{1}{k} \frac{2^k}{2^k - 1} \]

Keep a Record of All Corrected Bits

- For each bifaction step in a block with different parity, first check if one of the halves consists only of already corrected bits. If this is the case, the parity of this half is equal to Alice and Bob, and that for the other half is different. We need not transmit parity of neither half.
- Same argument for initialization of a new pass: If a block consists only of previously corrected bits, its parity must be the same for Alice and Bob, so it need not be exchanged.

5 IMPROVING THROUGHPUT

Different variants of the cascade protocol were tested. The most efficient variant (var. 1, see Box 6) was modified to achieve higher throughput at the cost of little lower efficiency. In var. 2 to (4), number of passes was 14. For var. 3 and (4), k = min \{2^k | 2^k - 1 > 2\} \cap \mathbb{N}. Other k values as for var. 1 (see Box 6).

6 SIMULATION RESULTS FOR MOST EFFICIENT METHOD (VAR. 1)

We use 16 passes, k_0 = \min(\{2^k | 2^k - 1 > 2\} \cap \mathbb{N}), k_{\text{th}} = 2^{-10}, k_{\text{th}} = n/2, and get a FER of 10^-4. Table and Fig. 7: Table: Block size n, cross-over probability p, efficiency \eta and throughput \eta and throughput in different cases. Efficiency values \eta, \epsilon and \eta', frame error rate \epsilon, bit error rate \epsilon', number of messages, throughput and latency. Figure 7: Efficiency, latency and throughput for different block sizes and cross-over probabilities on a logarithmic scale. 

References
