An Accurate Analysis of the BINARY Information Reconciliation Protocol by Generating Functions

SEAN SEET (NUS HIGH SCHOOL OF MATHEMATICS)

RUTH II-YUNG NG (UNIVERSITY OF CHICAGO)

KHOONGMING KHOO (DSO NATIONAL LABORATORIES)

Information Reconciliation (IR)

- Errors in raw key shared by Alice and Bob in quantum key exchange because of:
  - Eve's eavesdropping
  - Quantum channel noise

- Need information reconciliation to correct shared key by public exchange of parity bits.

- Want to:
  - Minimize leakage of parity bits
  - Maximize Decoding Success Probability (probability of having no errors left after decoding)

BINARY and CASCADE IR Protocol

- BINARY IR Protocol:
  1. At pass i, divide shared secret into blocks of length k_i
  2. Binary search error correction performed on each odd-parity block. Correct one error by exchanging $\log_2(k_i)$ parity bits.
  3. Not all errors corrected, so partially corrected secret is permuted and chopped up into blocks of size $k_{i+1} = 2k_i$.
  4. Repeat steps 2 to 3 for a number of passes, e.g. four or five passes.

- CASCADE IR Protocol:
  - In step 2, when an error bit is identified in a current pass, backtrack to correct more error bits which are paired with this bit in previous passes.

Objective

- To find the exact error probability distribution for the BINARY IR protocol at each pass.

- This will help us determine:
  - Leakage of parity bits
  - Decoding Success Probability

  for the BINARY protocol.

- Will help us in analysis of CASCADE, one of the popular IR protocol in use today.

Notations

Symbol	Meaning
n	Length of secret
k_i	Size of blocks at pass I
p	Bit error rate (BER)
$\Delta_i(j-y\|j)$	Probability that at i^{th} pass, j-y errors are corrected conditioned on there being j errors
$P_i(y)$	Probability that there are y errors at the i^{th} pass

Probability Distribution of BINARY

- **Theorem 1:** Let the initial probability distribution before BINARY IR is given by:

$$P_0(y) = \binom{n}{y} p^y (1-p)^{n-y}$$

The probability that there are y errors left in the raw key after pass i of the BINARY IR protocol is:

$$P_i(y) = \sum_{j=y}^{y+n/k_i} P_{i-1}(y)\Delta_i(j-y \mid y)$$

Proof Idea: Supposed there are j errors before executing pass i, we need to correct j-y errors to be left with y errors after pass i. Then we sum over all possible number of errors j before pass i.

Probability Distribution of BINARY

- **Theorem 2:** The quantity $\Delta_i(j\text{-}y|j)$ needed in Theorem 1 is computed by: $$\frac{\binom{n/k_i}{j-y}}{\binom{n}{j}} \times C_{i,j,y}$$

where $C_{i,j,y}$ is the coefficient of x^j in:

$$\left(\frac{(1+\mathrm{x})^{\mathrm{k_i}} - (1-x)^{k_i}}{2}\right)^{j-y} \left(\frac{(1+\mathrm{x})^{\mathrm{k_i}} + (1-x)^{k_i}}{2}\right)^{n-(j-y)}$$

Proof Idea: To correct j-y out of j errors, we need to distribute j error bits among n/k_i blocks such that j-y of them has odd parity.

Example

An example computation of Theorem 1 and comparison with simulation for n=2048-bit, BER = 3%.

	Pass $i=1$	Pass $i=2$	Pass $i=3$	Pass $i=4$
$P_i(0)$	0.00002	0.08808	0.63320	0.92559
$P_i(2)$	0.00020	0.17874	0.23883	0.06221
$P_i(4)$	0.00114	0.21261	0.08548	0.00974
$P_i(6)$	0.00421	0.19004	0.02891	0.00192
$P_i(8)$	0.01166	0.14029	0.00937	0.00042
$P_i(0)$	0.00000	0.08700	0.63740	0.92750
$P_i(2)$	0.00010	0.17690	0.23460	0.05950
$P_i(4)$	0.00110	0.21610	0.08540	0.01030
$P_i(6)$	0.00360	0.19370	0.02840	0.00230
$P_i(8)$	0.01070	0.13500	0.01020	0.00020

TABLE II
BINARY SIMULATION (TOP) AND CALCULATION (BOTTOM): $k_1 = 16$, $n = 2048$, $p = 0.03$

Brassard-Savail's Bound

- Brassard and Savail derived a bound to track the number of errors remaining in (an initial) block after pass i.
- They use $\delta_i(j)$ to denote the probability that there are 2j error bits remaining after pass i.
- They proved that $\delta_i(j) \leq \delta_{i-1}(j)/2$ under suitable conditions. From this, we get lower bound for decoding success probability:

$$P_i(0) \geq (1 - \sum_{i\neq 0}\delta_1(j)/2^{i-1})^{n/k1}$$

Comparison with Brassard-Savail's Bound

- Here we compare decoding success probability using our formula for BINARY and Brassard's lower bound for CASACADE:

Value	Our Calculation for BINARY	Brassard's Lower Bound for CASCADE
$P_1(0)$	0.00002	0.00002
$P_2(0)$	0.08808	0.00476
$P_3(0)$	0.63320	0.07106
$P_4(0)$	0.92559	0.26839
$P_5(0)$	0.98420	0.51894
$P_6(0)$	0.99492	0.72068
$P_7(0)$	0.99722	0.84902

TABLE VI

COMPARISON, $k_1 = 16$, $n = 2048$, $p = 0.03$

Comparison with Brassard-Savail's Bound

- CASCADE has better error correction performance than BINARY because of the backtracking step.
- So decoding success probability for CASCADE should be higher than BINARY.
- From the comparison table in previous slide, we see that the lower bound of decoding success from Brassard-Savail's formula may not be tight enough.

Conclusion

- Brassard-Savail's formula
  - Focuses on bound for error probability distribution within a block for CASCADE IR.
  - Is very easy to compute.
- But when extrapolated to deduce behavior of error distribution across the whole string, it may not be tight enough.
- Our formula
  - Calculates exact probability for BINARY IR. Hope to extend this formulation to compute probability for CASCADE.
  - Can be complex to compute for large n. Hope to find an efficient way to comupte the generating functions coefficients $C_{i,j,y}$