A Family of Finite Geometry LDPC Codes for Quantum Key Expansion

Kung-Chuan Hsu, Todd A. Brun
kungchuh@usc.edu, tbrun@usc.edu
Communication Sciences Institute, University of Southern California

Motivation

- A quantum key distribution (QKD) protocol allows two parties Alice and Bob to establish secret key through one-way quantum communications and classical public communications. With the aid of an entanglement-assisted (EA) quantum error correcting (QEC) code, QKD can be made to expand a secret key, and the process is known as quantum key expansion (QKE).
- Luo and Devetak[1] derived a QKE protocol by modifying the EA entanglement-distillation (EAED) protocol with the use of EA Calderbank-Shor-Steane (CSS) QEC codes.
- Our initial goal is to search for QEC codes that guarantee high net key rates for QKE.

The Original QKE Protocol

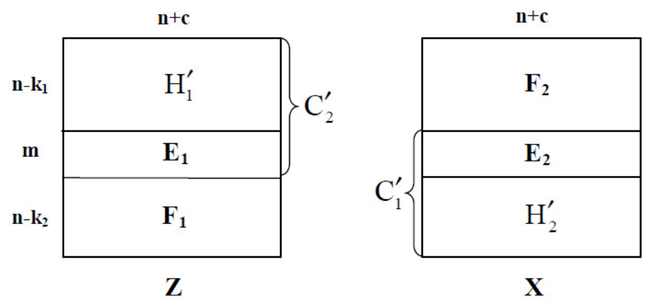
- Let C_1 and C_2 be two classical $[n, k_1]$ and $[n, k_2]$ codes with parity-check matrices $\boldsymbol{H_1}$ and $\boldsymbol{H_2}$.
- Let $\boldsymbol{M} = \boldsymbol{H_1}\boldsymbol{H_2^T}$ and $c = rank(\boldsymbol{M})$, $\exists$ nonsingular matrices $\boldsymbol{T_1}$ and $\boldsymbol{T_2}$ s.t.

$$\boldsymbol{T_1 M T_2^T} = \begin{pmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I_c} \end{pmatrix}$$

- For $i = 1, 2$, let

$$\boldsymbol{H_i'} = (\boldsymbol{T_i H_i}, \boldsymbol{J_i}), \text{ where } \boldsymbol{J_i} = \begin{pmatrix} \boldsymbol{0_{(n-k_i-c)\times c}} \\ \boldsymbol{I_c} \end{pmatrix}$$

- Since $\boldsymbol{H_1' H_2'^T} = \boldsymbol{0}$, a $[[n, k_1 + k_2 - n + c; c]]$ CSS EAQEC code can be constructed from C_1' and C_2' whose parity-check matrices are $\boldsymbol{H_1'}$ and $\boldsymbol{H_2'}$.
- $\exists$ full rank matrices $\boldsymbol{E_1}, \boldsymbol{F_1}, \boldsymbol{E_2}, \boldsymbol{F_2}$ s.t.



where the product of the left matrix and the transpose of the right matrix is the identity.

- Based on the decoder and the channel, the error set correctable by the code C_1' is

$$\mathcal{E}_1' = \{\boldsymbol{F_2^T}\,\underline{s} \oplus \boldsymbol{E_2^T}\,\underline{\beta(s)} \oplus \boldsymbol{H_2'^T}\,\underline{\beta'(s)} : \underline{s} \in \mathbb{Z}_2^{n-k_1}\} \quad (1)$$

where $\underline{\beta(\cdot)} : \mathbb{Z}_2^{n-k_1} \to \mathbb{Z}_2^m$, $\underline{\beta'(\cdot)} : \mathbb{Z}_2^{n-k_1} \to \mathbb{Z}_2^{n-k_2}$.

- **Luo and Devetek's QKE protocol:**
  (1) Alice generates random bit string $\underline{a}$ and $\underline{\alpha}$ of length $(2 + 3\delta)n$. She prepares each bit in $\underline{a}$ in the Z or X basis according to the corresponding bit in $\underline{\alpha}$.
  (2) Alice sends the prepared qubits to Bob.
  (3) Bob generates a random $\underline{\gamma}$ of length $(2 + 3\delta)n$. He measures the qubits in the basis according to $\underline{\gamma}$, resulting in $\underline{b}$.
  (4) Alice announces $\underline{\alpha}$. Bob discards the bits in $\underline{b}$ where the corresponding bits in $\underline{\gamma}$ and $\underline{\alpha}$ don't match. With high probability, at least $(1 + \delta)n$ bits are left; if not, they abort the protocol.
  (5) Alice randomly chooses n bits from the remaining bit-string and announces the bit locations for Bob to extract the corresponding bits, resulting in $\underline{\hat{a}}$ and $\underline{\hat{b}}$. The rest are for channel estimation.
  (6) Let $\underline{\kappa}$ be the length-c pre-shared key. Alice computes $\underline{s_A} = \boldsymbol{H_1'} \begin{pmatrix} \underline{\hat{a}} \\ \underline{\kappa} \end{pmatrix}$ and announces to Bob.
  (7) Bob computes $\underline{s_B} = \boldsymbol{H_1'} \begin{pmatrix} \underline{\hat{b}} \\ \underline{\kappa} \end{pmatrix}$, and his part of the generated key is $\underline{k_B} = \boldsymbol{E_1} \begin{pmatrix} \underline{\hat{b}} \\ \underline{\kappa} \end{pmatrix} \oplus \beta(\underline{s_A} \oplus \underline{s_B})$.
  (8) Alice computes her key as $\underline{k_A} = \boldsymbol{E_1} \begin{pmatrix} \underline{\hat{a}} \\ \underline{\kappa} \end{pmatrix}$.

- The post-processing stage consists of steps **(4)**-**(8)**.
- The net key rate of QKE is $\frac{m-c}{n}$

Contribution

- We formalize the error correction steps in the post-processing stage which are not described in detail in the original QKE protocol.
- We propose an improved QKE protocol based on the observation and fact that block error rate of the codes affect much the bit error performance of QKE, and to ensure the bit error threshold is met.
- We investigated in a family of finite geometry LDPC codes for use in QKE. We select codes from the family that have the best QKE performances in various channel error regions.

Formalism of Error Correction in QKE for LDPC Codes

- $\underline{\beta(\cdot)}$ in step **(7)** acts as the error recovery process considering the syndrome $\underline{s_A} \oplus \underline{s_B}$.
- The LDPC codes that we consider perform well classically. In order to utilize the power of these LDPC codes, the Sum-Product-Algorithm (SPA) decoder is considered. Furthermore, Bob may take advantage of the power of the original LDPC code by decoding using the original LDPC matrix $\boldsymbol{H_1}$ instead of $\boldsymbol{H_1'}$. This is possible since the last c bits of the message are error-free, and the syndrome w.r.t. $\boldsymbol{H_1}$ can be retrieved from the syndrome w.r.t. $\boldsymbol{H_1'}$.

The Improved QKE Protocol

- We modified the QKE protocol starting from step **(6)** as follows:
  (6) Alice computes $\underline{s_A} = \boldsymbol{H_1'} \begin{pmatrix} \underline{\hat{a}} \\ \underline{0_c} \end{pmatrix}$ and announces to Bob.
  (7) Let $\underline{s_B} = \boldsymbol{H_1'} \begin{pmatrix} \underline{\hat{b}} \\ \underline{0_c} \end{pmatrix}$. Bob computes the estimated error, $\underline{\hat{e}}$, by decoding the syndrome $\boldsymbol{T_1^{-1}}(\underline{\boldsymbol{s_A}} \oplus \underline{\boldsymbol{s_B}})$ w.r.t. $\boldsymbol{H_1}$.
  (8) Bob checks if the syndrome of $\underline{\hat{e}}$ w.r.t. $\boldsymbol{H_1}$ is $\underline{0}$; if not, they abort the protocol.
  (9) Bob randomly chooses μ bits from $\underline{k'_B} = \boldsymbol{E_1} \begin{pmatrix} \underline{\hat{b}} \oplus \underline{\hat{e}} \\ \underline{0_c} \end{pmatrix}$ and announces to Alice. Alice compares them with the corresponding bits from $\underline{k'_A} = \boldsymbol{E_1} \begin{pmatrix} \underline{\hat{a}} \\ \underline{0_c} \end{pmatrix}$. If there is a mismatch, they abort the protocol.
  (10) Alice computes her key $\underline{k_A} = \underline{k'_A} \oplus \boldsymbol{E_1} \begin{pmatrix} \underline{0_n} \\ \underline{\kappa} \end{pmatrix}$. Bob computes his key $\underline{k_B} = \underline{k'_B} \oplus \boldsymbol{E_1} \begin{pmatrix} \underline{0_n} \\ \underline{\kappa} \end{pmatrix}$.
  (11) Alice randomly discards μ bits from $\underline{k_A}$, and she announces the bit locations for Bob to discard the corresponding bits in $\underline{k_B}$.
- μ is to be chosen so that the generated keys have BER$< \epsilon$. Let R_{Blk} be the block error rate of the original QKE. Let p_1 be rate of abort at step **(8)**. Conditioned on surviving **(8)**, let p_2 be the BER in the remaining block errors.

$$\mu = \begin{cases} \lceil log_{(1-p_2)}(\frac{\epsilon(1-R_{blk})}{(p_2-\epsilon)(R_{blk}-p_1)}) \rceil & \text{, if } p_2 > \epsilon \\ 0 & \text{, otherwise} \end{cases}$$

- The net key rate of this improved QKE protocol is $R_{net} = (1 - R_{blk} + (1 - p_2)^{\mu}\,(R_{blk} - p_1))\,\frac{m-c-\mu}{n}$.

Finite Geometry LDPC Codes

- A geometric approach in constructing LDPC codes was presented[2], and the family was named "finite geometry".
- Two constructions of FG LDPC codes are Euclidean geometry(EG) and projective geometry(PG). The type of construction together with two parameters p and s specify a code.
- Can be extended by performing "column and row-splitting" on the parity-check matrices with pair of factors (c_{sp},r_{sp}).

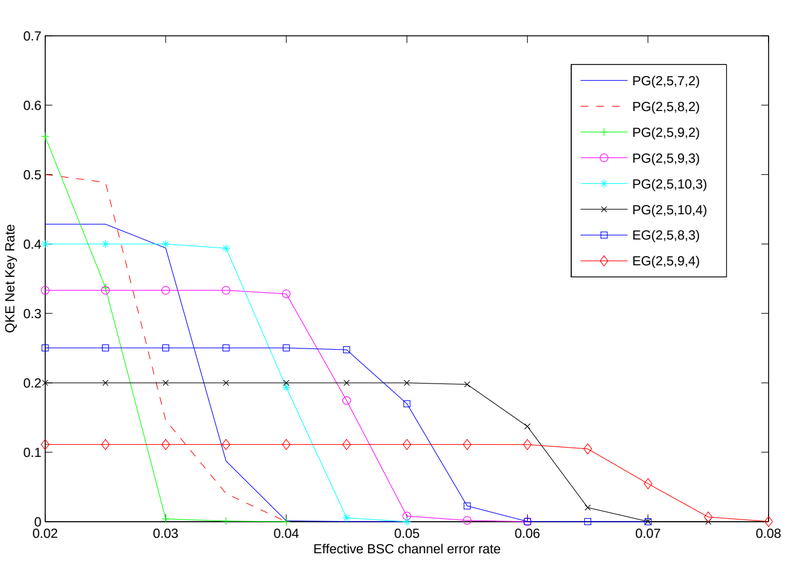
QKE Simulation and Result

Method: Monte Carlo simulation with 200,000 samples
Channel Model: Depolarizing channel (effectively a BSC)
Decoder: SPA decoder with maximum iteration =100
Codes: Extended FG with $(p, s) = (2, 5)$ and $n \leq 11,000$
Error Threshold: $\epsilon = 10^{-6}$

For effective BSC error rate $\leq$ 0.01, use $EG(2, 5, 10, 2)$, which is a $[[10230, 8182; 2044]]$ code with $R_{net} = 0.6$
For effective BSC error rate $\in (0.01, 0.02)$, use $EG(2, 5, 9, 2)$.
For effective BSC error rate $\geq$ 0.02, refer to the figure below.

There are 48 codes in the family with net key rate $R_{net} > 0$. Selected codes for effective BSC error rate $\geq 2\%$ are shown in the table. They outperform other codes in the family in specific channel error regions.

$[[n, m; c]]$	$FG(p, s, c_{sp}, r_{sp})$	R_{net}
$[[7399, 5285; 2114]]$	$PG(2, 5, 7, 2)$	0.4286
$[[8184, 5115; 3067]]$	$EG(2, 5, 8, 3)$	0.2502
$[[8456, 6342; 2112]]$	$PG(2, 5, 8, 2)$	0.5002
$[[9207, 5115; 4092]]$	$EG(2, 5, 9, 4)$	0.1111
$[[9513, 7399; 2114]]$	$PG(2, 5, 9, 2)$	0.5556
$[[9513, 6342; 3171]]$	$PG(2, 5, 9, 3)$	0.3333
$[[10570, 7399; 3171]]$	$PG(2, 5, 10, 3)$	0.4000
$[[10570, 6342; 4228]]$	$PG(2, 5, 10, 4)$	0.2000



References

[1] Z. Luo, I. Devetak, "Efficiently Implementable Codes for QUantum Key Expansion," *Phys. Rev. A*, vol. 75, Jan. 2007.

[2] Y. Kou, S. Lin, M. Fossorier, "Low-Density Parity-Check Codes Based on Finite Geometries: A Rediscovery and New Results," *IEEE Trans. Inform. Theory*, vol. 47, no. 7, pp. 2711-2736, 2001.