In this paper, we propose a construction for locally repairable codes (LRCs) with locality 2 using the concatenated structure of the punctured code and cyclic code. From the proposed construction, we found some new LRCs with larger dimension then those by C. Kim in 2018 at some length of \( n \).

### Introduction
- The locality \( r \) of the locally repairable codes (LRCs), which is the number of symbols needed to repair a symbol in a failed node.
- [1] proposed a construction of binary LRCs with \( r = 2 \) using the concatenated structure of the shortened code.

### Motivation
- The redundancy of code remains after the shortening, and the redundancy increases after the puncturing.
- We substitute the punctured code for the shortened code.

### Main construction
- Let \( n \) be an integer with \( n \geq 9 \) and \( 3 \mid n \);
- Let \( u \) be an odd positive integer and \( v \) be any positive integer, which satisfy \( uv = \frac{2n}{3} + 1 \) or \( \frac{2n}{3} \), and \( \gcd(u, v) = 1 \);
- Let \( \beta \) be a primitive \( u^v \) root of unity in some extension field of \( F_2 \) and \( C_c \) be a binary \([uv, v(u - 1) - \deg(g_1(x))] \) cyclic code with generator polynomial \( g(x) = (x^v + 1)g_1(x) \), where \( g_1(x) \) is the minimal polynomial of \( \beta \) over \( F_2 \);
- Then, a binary \([n, v(u - 1) - \deg(g_1(x)), d \geq 4, 2]\) LRC is constructed using the concatenated structure of the following outer code \( C_{\text{out}} \) and inner code \( C_{\text{in}} \):
  - \( C_{\text{out}} \): a binary code is generated by puncturing the last \((uv \mod 2)\) bit of \( C_c \).
  - \( C_{\text{in}} \): a binary \([n, \frac{2n}{3}] \) cyclic code with parity check polynomial \( h(x) = x^{20} + x^{10} + 1 \).

### Example:
- Let \( n = 30 \).
- We can choose \( v = 3 \), \( u = 7 \) and \( g_1(x) = x^3 + x + 1 \). And we get a binary \([21, 15]\) cyclic code \( C_c \) with generator polynomial \( g(x) = (x^3 + 1)(x^3 + x + 1) \).
- The outer code \( C_o \) and inner code \( C_{\text{in}} \) are used in the concatenated structure is defined as:
  - \( C_o \): a binary \([20, 15]\) punctured code by puncturing the last bit of \( C_c \).
  - \( C_{\text{i}} \): a binary \([30, 20]\) cyclic code with parity check polynomial \( h(x) = x^{20} + x^{10} + 1 \).

Therefore, we get a binary \([30, 15, \geq 4, 2]\) LRC.

### Some constructed LRCs

<table>
<thead>
<tr>
<th>Proposed construction</th>
<th>(9, 3)</th>
<th>(21, 10)</th>
<th>(30, 15)</th>
<th>(45, 25)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Construction in [1]</td>
<td>(9, 2)</td>
<td>(21, 9)</td>
<td>(30, 14)</td>
<td>(45, 24)</td>
</tr>
</tbody>
</table>

### Conclusion
- We propose a new construction for binary LRCs and found some new LRCs with larger dimension at some length of \( n \).
- In the future, it is necessary to find other outer code to increase the dimension at the given length of \( n \).

### REFERENCES