Q1 Teamname

0 Points

NULL

Q2 Commands

5 Points

List the commands used in the game to reach the ciphertext.

go,wave,dive,go,read,password

Q3 Analysis

50 Points

Give a detailed description of the cryptanalysis used to figure out the password. (Explain in less than 100 lines and use Latex wherever required. If your solution is not readable, you will lose marks. If necessary, the file upload option in this question must be used TO SHARE IMAGES ONLY.)

1. Finding the Encoding of the alphabet

Firstly, we tried finding the encoding used for the alphabet by giving "password" as input and then again the obtained encrypted text as input and so on. On analyzing the output it seemed that the encoding is the same as in the previous assignment that is, letters from 'f' to 'u' correspond to 0 to 15. However, unlike the previous assignment since each byte is interpreted as an element from F_{128} , each byte can take only values from "ff"(0) to "mu"(127). To verify that our claims were true, we used the program "analyze_encoding.cpp" and concluded that our observations were true.

2. Observing that A is lower triangular

The next main observation was that when we give inputs in which a prefix of bytes are zero, we got outputs whose corresponding prefix were also zeroes. Also changing a particular byte in the input only changes the output in the bytes after the changed byte, the bytes before the changed byte remained the same. This made us realize that matrix A might be lower triangular. So we

assumed that A is lower triangular for the rest of the analysis and we were able to find the password. To analyze A, we used the program "analyze_A.cpp".

3. Finding Diagonal elements of A and elements of E

Since A is lower triangular when we give inputs in which only the ith byte is non-zero, the ith byte of output would be $\left(A_{ii}\left(A_{ii}B_i^{E_i}\right)^{E_i}\right)^{E_i}$, where A_{ii} is the ith element in the main diagonal of A, B_i is the value of ith byte in the input (which is the only non-zero byte in the input) and E_i is the value of ith byte of E. So we passed all possible inputs (0 to 127) as B_i and obtained their outputs and then bruteforced the values for A_{ii} (0 to 127) and E_i (1 to 126) and found the values which mapped all the inputs to their corresponding outputs. After doing this for all $1 \leq i \leq 8$, we obtained a set of possible pairs of values A_{ii} for each A_{ii} which are given below. The code for this can be found in "solver.cpp" (lines 140-169).

For i=1: (8,41) (84,17) (109,69)

For i=2: (70,108) (77,20)

For i=3: (43,36) (78,42) (87,49)

For i = 4: (12,72) (75,84) (105,98)

For i=5: (47,65) (96,97) (112,92)

For i=6: (11,53) (41,83) (127,118)

For i=7: (14,108) (27,20)

For i=8: (38,15) (61,31) (125,81)

4. Using the above findings to find other elements of A.

Since we now have the diagonal elements of A and elements of E, we can use them to find the remaining elements of A. Our Strategy was to use the inputs in which jth byte is non-zero and analyze the ith bit of the corresponding output to find the value of A_{ij} . The ith byte of output when jth byte of input is the only non-zero byte in the input is given by

$$O_i = \left(\sum_{k=j}^{k=i} A_{ik} (A_{kj} B_j^{E_j})^{E_k}
ight)^{E_i}$$

As we can see, to obtain A_{ij} from the above equation, we need to need to already have the values of A_{ik} and A_{kj} where j < k < i along with diagonal elements. So inorder to satisfy these dependencies we compute

values of A_{ij} in increasing order of i and for each i we find them in decreasing order of j. This way, we would have found all dependencies required for A_{ij} before computing it. Finally to find A_{ij} we can brute force all values for A_{ij} from 0 to 127 and find values which map all the 127 possible inputs (with only jth byte non-zero) to their corresponding outputs. Although there are multiple possibilities for diagonal elements as shown above, some of them do not produce valid A_{ij} for some i,j, So we eliminated them and finally were left with unique entries for all positions in the matrix. The code for this can be found in "solver.cpp" (lines 171-228). The matrix A and E obtained are as follows:

$$A = \begin{pmatrix} 84 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 125 & 70 & 0 & 0 & 0 & 0 & 0 & 0 \\ 20 & 16 & 43 & 0 & 0 & 0 & 0 & 0 \\ 101 & 16 & 25 & 12 & 0 & 0 & 0 & 0 \\ 111 & 43 & 6 & 122 & 112 & 0 & 0 & 0 \\ 28 & 46 & 30 & 33 & 110 & 11 & 0 & 0 \\ 9 & 118 & 14 & 105 & 26 & 88 & 27 & 0 \\ 89 & 3 & 95 & 28 & 24 & 71 & 2 & 38 \end{pmatrix}$$

$$E = (17\ 108\ 36\ 72\ 92\ 53\ 20\ 15)$$

5. Decrypting the password.

The encrypted password was given to be

"immlmokkmgisjsmljufkhkimhmflgkgp". We converted into bits using the encoding in section 1. The password is of 16 bytes length. So we divided into two parts of 8 bytes each. To find the corresponding input for each 8 byte part, we did the following: we processed each byte sequentially from first to eigth and tried all possible values(0 to 127) for that byte and selected the values for which the output bytes after encryption(EAEAE) match with the given value upto that byte. Since ith byte of output depends only on bytes $\leq i$ in the input, we would end up with correct values using the above procedure since we process bytes from i=1 to i=8. The code for this can also be found in "solver.cpp"(lines 230-290). We used ascii representation like the previous assignment to convert from bit representation to alphabetic representation and got the following password "qmmndzmlga000000". Similar to previous assignment, removing zeroes at the end clears the level.

So the password for Level 5 is "qmmndzmlga".

No files uploaded

Q4 Password

5 Points

What was the final command used to clear this level?

```
qmmndzmlga
```

Q5 Codes

0 Points

```
♣ Download
▼ analyse_encoding.cpp
 1
     #include <bits/stdc++.h>
 2
     using namespace std;
 3
 4
    // to find the encoding of the alphabet
 5
    mt19937
     rng(chrono::steady clock::now().time since epoch().count());
 7
     int getRand(int l, int r)
 8
     {
 9
         uniform int distribution<int> uid(l, r);
         return uid(rng);
10
11
     }
12
13
     void getinputs()
14
15
         ofstream fout("test_inputs.txt");
16
17
         int L = 1000;
18
         for (int i = 0; i < L; i++)
19
20
             for (int j = 0; j < 8; j++)
21
22
                  int k = getRand(0, 7);
                 fout << char('f' + k);</pre>
23
24
                  k = getRand(0, 15);
25
                  fout << char('f' + k);</pre>
26
             }
27
             fout << endl;
28
         }
```

```
29
    }
30
31
    void getoutputs()
32
        ifstream fin("test inputs.txt");
33
34
        ofstream fout("test cmds.txt");
35
36
        fout << "NULL" << endl;
37
        fout << "foobar268" << endl;</pre>
38
        fout << 5 << endl;
39
        fout << "go" << endl;</pre>
40
41
        fout << "wave" << endl;
42
        fout << "dive" << endl;</pre>
43
        fout << "go" << endl;
44
        fout << "read" << endl;</pre>
45
46
        string s;
47
        while (fin >> s)
48
        {
49
             fout << s << endl;
            fout << 'c' << endl;
50
51
        }
52
53
        fout << "back" << endl;</pre>
54
        fout << "exit" << endl;
55
56
        fout.close();
57
        fin.close();
58
59
        system("ssh student@65.0.124.36 < test cmds.txt > out");
60
        system("grep --no-group-separator -A 1 \"Slowly, a new text
    starts appearing on the screen. It reads ...\" out | grep --no-
    group-separator -v \"Slowly, a new text starts appearing on the
    screen. It reads ...\" | tr -d \"\\t\" > test outputs.txt");
61
        system("rm -rf out test cmds.txt");
62
    }
63
64
    void analyze()
65
        ifstream fin("test_outputs.txt");
66
        vector<int> cnt(26);
67
68
69
        string s;
70
        fin >> s;
71
        while (fin >> s)
72
73
            for (char c : s)
74
                 cnt[c - 'a']++;
75
        }
76
77
        for (int i = 0; i < 26; i++)
```

```
78
         {
79
             cout << char('a' + i) << ' ' << cnt[i] << endl;</pre>
80
         cout << endl;</pre>
81
82
    }
83
    int main()
84
85
    {
86
         getinputs();
87
         getoutputs();
88
         analyze();
89
         return 0;
90
91
    }
92
```

♣ Download

```
▼ analyse_A.cpp
 1
     #include <bits/stdc++.h>
 2
     using namespace std;
 3
 4
     // to check whether A is lower triangular.
 5
 6
     mt19937
     rng(chrono::steady clock::now().time since epoch().count());
 7
     int getRand(int l, int r)
 8
         uniform_int_distribution<int> uid(l, r);
 9
10
         return uid(rng);
11
12
13
     void getinputs()
14
15
         ofstream fout("test_inputs.txt");
16
17
         for (int i = 1; i \le 8; i++)
18
         {
19
             for (int j = 0; j < i; j++)
20
             {
21
                 fout << "ff";
22
             }
23
             for (int j = i; j < 8; j++)
24
             {
25
                 int k = getRand(0, 7);
26
                 fout << char('f' + k);</pre>
27
                 k = getRand(0, 15);
                 fout << char('f' + k);</pre>
28
29
             }
30
             fout << endl;
31
         }
32
33
```

```
34
    void getoutputs()
35
36
        ifstream fin("test inputs.txt");
37
        ofstream fout("test cmds.txt");
38
39
        fout << "NULL" << endl;
        fout << "foobar268" << endl;</pre>
40
        fout << 5 << endl;
41
42
        fout << "go" << endl;</pre>
43
44
        fout << "wave" << endl;
45
        fout << "dive" << endl;
46
        fout << "go" << endl;
47
        fout << "read" << endl;</pre>
48
49
        string s;
50
        while (fin >> s)
51
        {
52
             fout << s << endl;
53
             fout << 'c' << endl;</pre>
54
        }
55
56
        fout << "back" << endl;</pre>
57
        fout << "exit" << endl;</pre>
58
59
        fout.close();
60
        fin.close();
61
        system("ssh student@65.0.124.36 < test cmds.txt > out");
62
63
        system("grep --no-group-separator -A 1 \"Slowly, a new text
    starts appearing on the screen. It reads ...\" out | grep --no-
    group-separator -v \"Slowly, a new text starts appearing on the
    screen. It reads ...\" | tr -d \"\\t\" > test outputs.txt");
        system("rm -rf out test cmds.txt");
64
65
    }
66
67
    void analyze()
68
69
        ifstream fin("test outputs.txt");
70
        vector<int> cnt(26);
71
72
        string s;
73
        fin >> s;
74
        int i = 1;
75
        bool ok = 1;
76
        while (fin >> s)
77
78
             for (int j = 0; j < 2 * i; j++)
79
             {
80
                 ok &= (s[j] == 'f');
81
             }
82
        }
```

```
83
84
        assert(ok);
85
86
87
    int main()
88
89
        getinputs();
90
        getoutputs();
91
        analyze();
92
93
        return 0;
94
    }
95
```

```
♣ Download
▼ solver.cpp
 1
     #include <bits/stdc++.h>
 2
     using namespace std;
 3
 4
    // breaks the cipher assuming A is lower triangular
 5
 6
     using b7 = bitset<7>;
 7
     using b13 = bitset<13>;
 8
 9
     #define endl '\n'
10
11
     b13 \mod = b13("10000011");
12
13
    b7 add(b7 x, b7 y)
14
15
         x ^= y;
16
         return x;
17
     }
18
     b7 rem(b13 x)
19
20
21
         for (int i = 12; i >= 7; i--)
22
23
             if (x[i] != 0)
24
             {
25
                 x ^= (mod << (i - 7));
26
             }
27
         }
28
29
         for (int i = 12; i \ge 7; i--)
30
         {
31
             assert(x[i] == 0);
32
         }
33
34
         b7 ans;
35
         for (int i = 0; i < 7; i++)
36
             ans[i] = x[i];
```

```
37
38
         return ans;
39
40
41
    b7 mul(b7 x, b7 y)
42
43
        b13 res;
44
         res.reset();
        for (int i = 0; i < 7; i++)
45
46
47
             for (int j = 0; j < 7; j++)
48
             {
49
                 if (x[i] == 1 \text{ and } y[j] == 1)
                     res[i + j] = res[i + j] ^ 1;
50
51
             }
52
53
        return rem(res);
54
    }
55
56
    b7 power(b7 a, int n)
57
58
        b7 res = 1;
59
        while (n)
60
61
             if (n & 1)
62
                 res = mul(res, a);
63
             n >>= 1;
64
             a = mul(a, a);
65
66
        return res;
67
68
69
    b7 convert(char a, char b)
70
        int foo = ((a - 'f') << 4) + (b - 'f');
71
72
        assert(foo < 128);
73
        b7 res = foo;
74
        return res;
75
76
77
    void geninput()
78
79
        ofstream fout("in.txt");
        for (int i = 0; i < 8; i++)
80
81
82
             for (int j = 1; j < 128; j++)
83
                 for (int r = 0; r < 8; r++)
84
85
                 {
86
                     if (r != i)
                          fout << "ff";
87
88
                     else
```

```
89
                      {
90
                          fout << char('f' + (j >> 4));
91
                          fout << char('f' + (j & 15));
92
                      }
93
                  }
94
                  fout << endl;</pre>
95
             }
96
         }
97
98
99
     void genoutput()
100
101
         ifstream fin("in.txt");
102
         ofstream fout("cmds.txt");
103
         fout << "NULL" << endl;</pre>
104
         fout << "foobar268" << endl;</pre>
105
         fout << 5 << endl:
106
107
         fout << "go" << endl;
108
109
         fout << "wave" << endl;
110
         fout << "dive" << endl;</pre>
         fout << "go" << endl;
111
112
         fout << "read" << endl;</pre>
113
114
         string s;
115
         while (fin >> s)
116
117
             fout << s << endl;
118
             fout << 'c' << endl;
119
         }
120
121
         fout << "back" << endl;
122
         fout << "exit" << endl;
123
124
         fout.close();
125
         fin.close();
126
127
         system("ssh student@65.0.124.36 < cmds.txt > out");
128
         system("grep --no-group-separator -A 1 \"Slowly, a new text
     starts appearing on the screen. It reads ...\" out | grep --no-
     group-separator -v \"Slowly, a new text starts appearing on the
     screen. It reads ...\" | tr -d \"\\t\" > out.txt");
129
         system("rm -rf out cmds.txt");
130
    }
131
132
    int main()
133
134
         ios_base::sync_with_stdio(false), cin.tie(nullptr);
135
136
137
         geninput();
```

```
138
         genoutput();
139
140
         ifstream fin("out.txt");
141
         vector<pair<int, int>> adj[8];
142
143
         string in[8][128];
144
145
         for (int i = 0; i < 8; i++)
146
147
             for (int j = 1; j < 128; j++)
148
                 fin >> in[i][j];
149
             for (int val = 0; val < 128; val++)
150
             {
151
                 for (int e = 1; e \le 126; e++)
152
                 {
153
                     bool ok = 1;
154
                     for (int j = 1; j < 128; j++)
155
                     {
156
                          string s = in[i][j];
157
                          b7 res = convert(s[2 * i], s[2 * i + 1]);
158
                          b7 expected = mul(val, power(j, e));
                          expected = mul(val, power(expected, e));
159
160
                          expected = power(expected, e);
161
                          ok &= (res == expected);
162
                          if (!ok)
163
                              break;
164
                     }
165
                     if (ok)
166
                          adj[i].push back({val, e});
167
                 }
168
             }
169
         }
170
171
         vector<int> E(8);
172
         vector<vector<int>>> A(8, vector<int>(8));
173
174
         for (int i = 1; i < 8; i++)
175
176
             for (auto [all, el] : adj[i - 1])
177
             {
178
                 for (auto [a22, e2] : adj[i])
179
                 {
                     for (int val = 0; val < 128; val++)
180
181
                     {
182
                          bool ok = 1;
183
                          for (int j = 1; j < 128; j++)
184
185
                              string s = in[i - 1][j];
186
                              b7 foo = power(add(mul(val, power(mul(a11,
     power(j, e1)), e1)), mul(a22, power(mul(val, power(j, e1)), e2))),
     e2);
187
                              b7 got = convert(s[2 * i], s[2 * i + 1]);
```

```
188
                              ok &= (foo == got);
189
                          }
190
                          if (ok)
191
                          {
192
                              A[i - 1][i - 1] = a11;
193
                              A[i][i] = a22;
194
                              E[i - 1] = e1;
195
                              E[i] = e2;
                              A[i][i - 1] = val;
196
197
                          }
198
                     }
199
                 }
200
             }
201
         }
202
         for (int i = 0; i < 8; i++)
203
204
             for (int j = i - 2; j >= 0; j - -)
205
206
             {
207
                  for (int val = 0; val < 128; val++)
208
209
                      A[i][j] = val;
210
                      bool ok = 1;
211
                      for (int x = 1; x < 128; x++)
212
                      {
213
                          b7 foo = 0;
214
                          for (int k = j; k \le i; k++)
215
216
                              foo = add(foo, mul(A[i][k], power(mul(A[k]
     [j], power(x, E[j])), E[k])));
217
218
                          foo = power(foo, E[i]);
219
                          b7 got = convert(in[j][x][2 * i], in[j][x][2 *
     i + 1]);
220
                          ok &= (foo == got);
221
                      }
222
                      if (ok)
223
                      {
224
                          break;
225
                      }
226
                 }
227
             }
228
         }
229
230
         string password = "immlmokkmgisjsmljufkhkimhmflgkgp";
231
232
         for (int i = 0; i < 2; i++)
233
234
             b7 out[8];
235
             for (int j = 16 * i; j < 16 * i + 16; j += 2)
236
             {
237
                 out[(j - 16 * i) / 2] = convert(password[j],
```

```
password[j + 1]);
238
             }
239
             b7 x[8];
240
             for (int r = 0; r < 8; r++)
241
             {
242
                  for (int val = 0; val < 128; val++)
243
                  {
244
                      b7 xx[8];
245
                      for (int k = 0; k < 8; k++)
246
                          xx[k] = x[k];
247
                      x[r] = val;
248
                      for (int _{-} = 0; _{-} < 2; _{++})
249
                      {
                          for (int j = 0; j < 8; j++)
250
251
                          {
252
                              x[j] = power(x[j], E[j]);
253
                          }
254
                          b7 y[8];
255
                          for (int j = 0; j < 8; j++)
256
                          {
257
                              y[j] = 0;
258
                              for (int k = 0; k < 8; k++)
259
260
                                   y[j] = add(y[j], mul(A[j][k], x[k]));
261
                              }
262
                          }
263
                          for (int j = 0; j < 8; j++)
264
265
                              x[j] = y[j];
266
                          }
267
                      }
268
                      for (int j = 0; j < 8; j++)
269
                      {
270
                          x[j] = power(x[j], E[j]);
271
                      }
272
                      bool ok = 1;
273
                      for (int j = 0; j <= r; j++)
274
275
                          ok &= (x[j] == out[j]);
276
                      }
277
                      for (int j = 0; j < 8; j++)
278
                      {
279
                          x[j] = xx[j];
280
                      }
281
                      if (ok)
282
                      {
283
                          x[r] = val;
284
                          break;
285
                      }
286
                  }
287
                  cout << char(x[r].to ullong());</pre>
288
             }
```

```
289 }
290 cout << endl;
291
292 return 0;
293 }
294
```

```
Assignment 5
                                                                                UNGRADED
GROUP
AJAY PRAJAPATI
A5 - SURYADEVARA SAI KRISHNA
A11 - GARIMELLA MOHAN RAGHU
View or edit group
TOTAL POINTS
- / 60 pts
QUESTION 1
                                                                                        0 pts
Teamname
QUESTION 2
                                                                                        5 pts
Commands
QUESTION 3
                                                                                       50 pts
Analysis
QUESTION 4
Password
                                                                                        5 pts
QUESTION 5
                                                                                        0 pts
Codes
```