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.)

We broke the cipher using a Chosen Plain text attack which is described in detail below:

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}, E_i) for each i 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

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

As we can see, to obtain A_{ij} from the above equation, we 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

analyse_encoding.cpp Download

```cpp
#include <bits/stdc++.h>
using namespace std;

// to find the encoding of the alphabet

mt19937
rng(chrono::steady_clock::now().time_since_epoch().count());
int getRand(int l, int r)
{
    uniform_int_distribution<int> uid(l, r);
    return uid(rng);
}

void getinputs()
{
    ofstream fout("test_inputs.txt");

    int L = 1000;
    for (int i = 0; i < L; i++)
    {
        for (int j = 0; j < 8; j++)
        {
            int k = getRand(0, 7);
            fout << char('f' + k);
            k = getRand(0, 15);
            fout << char('f' + k);
        }
        fout << endl;
    }
}

void getoutputs()
{
    ifstream fin("test_inputs.txt");
    ofstream fout("test_cmds.txt");

    fout << "NULL" << endl;
    fout << "foobar268" << endl;
    fout << 5 << endl;

    fout << "go" << endl;
    fout << "wave" << endl;
    fout << "dive" << endl;
    fout << "go" << endl;
    fout << "read" << endl;

    string s;
    while (fin >> s)
    {
        fout << s << endl;
        fout << 'c' << endl;
    }

    fout << "back" << endl;
    fout << "exit" << endl;

    fout.close();
    fin.close();

    system("ssh student@65.0.124.36 < test_cmds.txt > out");
    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");
}

void analyze()
{
    ifstream fin("test_outputs.txt");
    vector<int> cnt(26);

    string s;
    fin >> s;
    while (fin >> s)
    {
        for (char c : s)
```

```cpp
            cnt[c - 'a']++;
    }

    for (int i = 0; i < 26; i++)
    {
        cout << char('a' + i) << ' ' << cnt[i] << endl;
    }
    cout << endl;
}

int main()
{
    getinputs();
    getoutputs();
    analyze();

    return 0;
}

```

analyse_A.cpp Download

```cpp
#include <bits/stdc++.h>
using namespace std;

// to check whether A is lower triangular.

mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
int getRand(int l, int r)
{
    uniform_int_distribution<int> uid(l, r);
    return uid(rng);
}

void getinputs()
{
    ofstream fout("test_inputs.txt");

    for (int i = 1; i <= 8; i++)
    {
        for (int j = 0; j < i; j++)
        {
            fout << "ff";
        }
        for (int j = i; j < 8; j++)
        {
            int k = getRand(0, 7);
            fout << char('f' + k);
            k = getRand(0, 15);
            fout << char('f' + k);
        }
        fout << endl;
    }
}

void getoutputs()
{
    ifstream fin("test_inputs.txt");
    ofstream fout("test_cmds.txt");

    fout << "NULL" << endl;
    fout << "foobar268" << endl;
    fout << 5 << endl;

    fout << "go" << endl;
    fout << "wave" << endl;
    fout << "dive" << endl;
    fout << "go" << endl;
    fout << "read" << endl;

    string s;
    while (fin >> s)
    {
        fout << s << endl;
        fout << 'c' << endl;
    }

    fout << "back" << endl;
    fout << "exit" << endl;

    fout.close();
    fin.close();

    system("ssh student@65.0.124.36 < test_cmds.txt > out");
    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");
}

void analyze()
{
    ifstream fin("test_outputs.txt");
    vector<int> cnt(26);

    string s;
    fin >> s;
    int i = 1;
    bool ok = 1;
    while (fin >> s)
    {
        for (int j = 0; j < 2 * i; j++)
        {
            ok &= (s[j] == 'f');
        }
    }

    assert(ok);
}

int main()
{
    getinputs();
    getoutputs();
    analyze();

    return 0;
}

```

solver.cpp   Download

```
#include <bits/stdc++.h>
using namespace std;

// breaks the cipher assuming A is lower triangular

using b7 = bitset<7>;
using b13 = bitset<13>;

#define endl '\n'

b13 mod = b13("10000011");

b7 add(b7 x, b7 y)
{
    x ^= y;
    return x;
}

b7 rem(b13 x)
{
    for (int i = 12; i >= 7; i--)
    {
        if (x[i] != 0)
        {
            x ^= (mod << (i - 7));
        }
    }

    for (int i = 12; i >= 7; i--)
    {
        assert(x[i] == 0);
    }

    b7 ans;
    for (int i = 0; i < 7; i++)
        ans[i] = x[i];

    return ans;
}

b7 mul(b7 x, b7 y)
{
    b13 res;
    res.reset();
    for (int i = 0; i < 7; i++)
    {
        for (int j = 0; j < 7; j++)
        {
            if (x[i] == 1 and y[j] == 1)
                res[i + j] = res[i + j] ^ 1;
        }
    }
    return rem(res);
}

b7 power(b7 a, int n)
{
    b7 res = 1;
    while (n)
    {
        if (n & 1)
            res = mul(res, a);
        n >>= 1;
        a = mul(a, a);
    }
    return res;
}

b7 convert(char a, char b)
{
    int foo = ((a - 'f') << 4) + (b - 'f');
    assert(foo < 128);
    b7 res = foo;
    return res;
}

void geninput()
{
    ofstream fout("in.txt");
    for (int i = 0; i < 8; i++)
    {
        for (int j = 1; j < 128; j++)
        {
            for (int r = 0; r < 8; r++)
            {
                if (r != i)
                    fout << "ff";
                else
                {
                    fout << char('f' + (j >> 4));
                    fout << char('f' + (j & 15));
                }
            }
            fout << endl;
        }
    }
}

void genoutput()
{
    ifstream fin("in.txt");
    ofstream fout("cmds.txt");

    fout << "NULL" << endl;
    fout << "foobar268" << endl;
    fout << 5 << endl;

    fout << "go" << endl;
    fout << "wave" << endl;
    fout << "dive" << endl;
    fout << "go" << endl;
    fout << "read" << endl;

    string s;
    while (fin >> s)
    {
        fout << s << endl;
        fout << 'c' << endl;
    }

    fout << "back" << endl;
    fout << "exit" << endl;

    fout.close();
    fin.close();

    system("ssh student@65.0.124.36 < cmds.txt > out");
    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");
    system("rm -rf out cmds.txt");
}

int main()
{
```

```cpp

    ios_base::sync_with_stdio(false), cin.tie(nullptr);

    geninput();
    genoutput();

    ifstream fin("out.txt");
    vector<pair<int, int>> adj[8];

    string in[8][128];

    for (int i = 0; i < 8; i++)
    {
        for (int j = 1; j < 128; j++)
            fin >> in[i][j];
        for (int val = 0; val < 128; val++)
        {
            for (int e = 1; e <= 126; e++)
            {
                bool ok = 1;
                for (int j = 1; j < 128; j++)
                {
                    string s = in[i][j];
                    b7 res = convert(s[2 * i], s[2 * i + 1]);
                    b7 expected = mul(val, power(j, e));
                    expected = mul(val, power(expected, e));
                    expected = power(expected, e);
                    ok &= (res == expected);
                    if (!ok)
                        break;
                }
                if (ok)
                    adj[i].push_back({val, e});
            }
        }
    }

    vector<int> E(8);
    vector<vector<int>> A(8, vector<int>(8));

    for (int i = 1; i < 8; i++)
    {
        for (auto [a11, e1] : adj[i - 1])
        {
            for (auto [a22, e2] : adj[i])
            {
                for (int val = 0; val < 128; val++)
                {
                    bool ok = 1;
                    for (int j = 1; j < 128; j++)
                    {
                        string s = in[i - 1][j];
                        b7 foo = power(add(mul(val, power(mul(a11, power(j, e1)), e1)), mul(a22, power(mul(val, power(j, e1)), e2))), e2);
                        b7 got = convert(s[2 * i], s[2 * i + 1]);
                        ok &= (foo == got);
                    }
                    if (ok)
                    {
                        A[i - 1][i - 1] = a11;
                        A[i][i] = a22;
                        E[i - 1] = e1;
                        E[i] = e2;
                        A[i][i - 1] = val;
                    }
                }
            }
        }
    }

    for (int i = 0; i < 8; i++)
    {
        for (int j = i - 2; j >= 0; j--)
        {
            for (int val = 0; val < 128; val++)
            {
                A[i][j] = val;
                bool ok = 1;
                for (int x = 1; x < 128; x++)
                {
                    b7 foo = 0;
                    for (int k = j; k <= i; k++)
                    {
                        foo = add(foo, mul(A[i][k], power(mul(A[k][j], power(x, E[j])), E[k])));
                    }
                    foo = power(foo, E[i]);
                    b7 got = convert(in[j][x][2 * i], in[j][x][2 * i + 1]);
                    ok &= (foo == got);
                }
                if (ok)
                {
                    break;
                }
            }
        }
    }

    string password = "immlmokkmgisjsmljufkhkimhmflgkgp";

    for (int i = 0; i < 2; i++)
    {
        b7 out[8];
        for (int j = 16 * i; j < 16 * i + 16; j += 2)
        {
            out[(j - 16 * i) / 2] = convert(password[j],
password[j + 1]);
        }
        b7 x[8];
        for (int r = 0; r < 8; r++)
        {
            for (int val = 0; val < 128; val++)
            {
                b7 xx[8];
                for (int k = 0; k < 8; k++)
                    xx[k] = x[k];
                x[r] = val;
                for (int _ = 0; _ < 2; _++)
                {
                    for (int j = 0; j < 8; j++)
                    {
                        x[j] = power(x[j], E[j]);
                    }
                    b7 y[8];
                    for (int j = 0; j < 8; j++)
                    {
                        y[j] = 0;
                        for (int k = 0; k < 8; k++)
                        {
                            y[j] = add(y[j], mul(A[j][k], x[k]));
                        }
                    }
                    for (int j = 0; j < 8; j++)
                    {
                        x[j] = y[j];
                    }
                }
                for (int j = 0; j < 8; j++)
                {
                    x[j] = power(x[j], E[j]);
                }
                bool ok = 1;
                for (int j = 0; j <= r; j++)
                {
                    ok &= (x[j] == out[j]);
                }
                for (int j = 0; j < 8; j++)
                {
                    x[j] = xx[j];
                }
                if (ok)
                {
                    x[r] = val;
                    break;
                }
            }
            cout << char(x[r].to_ullong());
        }
    }
    cout << endl;

    return 0;
}

```

Assignment 5

● UNGRADED

GROUP

AJAY PRAJAPATI
A5 - SURYADEVARA SAI KRISHNA
A11 - GARIMELLA MOHAN RAGHU
View or edit group

TOTAL POINTS

- / 60 pts

QUESTION 1

Teamname 0 pts

QUESTION 2

Commands 5 pts

QUESTION 3

Analysis 50 pts

QUESTION 4

Password 5 pts

QUESTION 5

Codes 0 pts