Week 3 - Stream and Block Ciphers

MAT260: Cryptology

Gary Hobson

Date May 19, 2025

Introduction:

The assigned computer problems this week are from Chapter 5, Problems 1, 3, and Chapter 6, Problem 1.

Make sure to review the example computer problems in "Appendix C.4 Examples for Chapter 5" and "Appendix C.5 Examples for Chapter 6" that work similar problems to those you are assigned.

Make sure to run your code so all relevant computations/results are displayed and then export your work as a PDF file for submission.

Chapter 5 Problems:

Problem 1:

```
% Define the sequence
seq = [1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, ...
        0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, ...
        0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1];

%Guess the length
lfsrlength(seq, 20)
```

```
Order   Determinant
1        1
2        1
3        1
4        0
5        1
6        1
7        0
8        0
9        0
10        0
11        0
12        0
13        0
14        0
15        2
16        4.5475e-13
17        2
18        2
```

```
19        5.8208e-11
20        9.4587e-11
```

```
coeffs = lfsrsolve(seq, 6)
```

```
coeffs = 1×6
     1     1     0     1     1     0
```

```
%The recurrence relation is:
fprintf('x(n+6) = ');
```

```
x(n+6) =
```

```
for i = 1:length(coeffs)
    if coeffs(i)
        fprintf(' + x(n+%d)', i-1);
    end
end
```

```
 + x(n+0) + x(n+1) + x(n+3) + x(n+4)
```

```
fprintf(' mod 2\n');
```

```
 mod 2
```

Problem 3:

```
% Known portion of the plaintext and ciphertext
plaintext_known = [1 0 0 1 0 0 1 0 0 1 0 0 1 0 0];
ciphertext = [0 1 1 0 0 0 1 0 1 0 1 1 1 0 0 1 1 1 0, ...
              1 0 1 0 0 0 1 0 0 0 1 1 0 0 0 1 0 1 0, ...
              1 1 1 0 0 1 1 1 0 1 0 1];

% Get LFSR keystream start
lfsr_start = mod(plaintext_known + ciphertext(1:length(plaintext_known)), 2);

% Guess LFSR length
lfsrlength(lfsr_start, 8);  % Visual inspection suggests length is 5
```

```
Order   Determinant
1        1
2        0
3        0
4        1
5        1
6        0
7        0
8        0
```

```
% Solve for recurrence coefficients
coeffs = lfsrsolve(lfsr_start, 5);

% Generate full LFSR sequence (same length as ciphertext)
keystream = lfsr(coeffs, lfsr_start(1:5), length(ciphertext));
```

```
% Recover full plaintext
plaintext_full = mod(ciphertext + keystream, 2);

% Recovered Plaintext
disp(plaintext_full);
```

```
     1     0     0     1     0     0     1     0     0     1     0     0     1     0     0     1     0
```

Chapter 6 Problems:

Problem 1:

```
% Define the encryption matrix M (given)
M = [6, 24, 1;
     13, 16, 10;
     20, 17, 15];

% Ensure M is invertible mod 26
d = round(det(M));
if gcd(d, 26) ~= 1
    error('Matrix is not invertible mod 26.');
end

% Function to compute modular inverse of a mod m
function inv = modinv(a, m)
    [g, x, ~] = gcd(a, m);
    if g ~= 1
        error('modinv:inverse Does Not Exist', 'No modular inverse exists');
    else
        inv = mod(x, m);
    end
end

% Compute modular inverse of the determinant mod 26
detM = round(det(M));
detInv = modinv(detM, 26);

% Compute adjugate matrix of M
adjM = round(detM * inv(M));

% Compute inverse of M mod 26
Minv = mod(detInv * adjM, 26);

% Convert ciphertext to numerical form
ciphertext = 'zirkzwopjjoptfapuhfhadrq';
nums = double(ciphertext) - double('a');

% Reshape into blocks of 3 letters
nums = reshape(nums, 3, [])';

% Decrypt each block using modular matrix multiplication
```

```
plaintext_nums = mod(nums * Minv', 26);

% Convert numbers back to letters
plaintext = char(plaintext_nums' + double('a'));
plaintext = plaintext(:)';  % Flatten
disp(plaintext);
```

```
ukdjokrfagsjvxreynxfhjpf
```