Q1

a)

```
Secret key: 11010100010001101010
Publicly known seed: 00110110010101101111
Plaintext: 00111101011100111000000111001100010011010110110100
Alice takes the seed and secret key and creates the LFSR
Alice runs the LFSR to generate a keystream:
00110110010101101111110101101111101111001000001101
Alice generated ciphertext: 00001011001001010111111101111011100100110010111001
Alice sends the ciphertext to Bob
Bob takes the seed and secret key and creates the LFSR
Bob runs the LFSR to generate a keystream:
00110110010101101111110101101111101111001000001101
Bob decrypts the ciphertext to get the plaintext:
00111101011100111000000111001100010011010110110100
```

b)

Trudy always has access to ciphertext since its easy to pick up messages which is why encryption was made. Since Trudy also has access to some of the keystream, she can set up a matrix to solve for the secret coefficients.

$$\begin{array}{llll}
i=0 & s_m & = p_{m-1}s_{m-1} + \cdots + p_1 s_1 + p_0 s_0 & \bmod 2 \\
i=1 & s_{m+1} & = p_{m-1}s_m + \cdots + p_1 s_2 + p_0 s_1 & \bmod 2 \\
\vdots & \vdots & \vdots & \vdots \\
i=m-1 & s_{2m-1} & = p_{m-1}s_{2m-2} + \cdots + p_1 s_m + p_0 s_{m-1} & \bmod 2
\end{array}$$

Assuming trudy has at least 2m keystream bits (where m is 20 in this scenario, thus trudy needs 40 keystream bits), she can set up a system of linear equations. In this system, all the s_x are known since those are the known keystream bits. We can solve for all the p values which would give us the key.

c)

In the case where Trudy has access to the plaintext, Trudy can still carry out the attack because Trudy can bitwise XOR the plaintext with the ciphertext to produce the keystream bits for that plaintext. As long as Trudy has access to 40 bits of plaintext, just like in part b, then trudy can use the keystream bits to perform the attack mentioned in part b.

2

Affine Cipher's are still vulnerable to frequency analysis attack. The mapping is 1 to 1, otherwise encrypting and the decrypting would not produce the same results. Each letter will be assigned a new element from Z_m . If we know the most frequent letter in the plaintext (typically by assuming it follows the known distribution of letters for the alphabet), then we can try different a,b values such that the most frequent letter in y would be the decrypted into the most frequent known letter in plaintext x. Attackers can also try this for multiple letters of high frequency.

For instance, in English the highest frequency letter is "E". A hacker can take a ciphertext and look at its most common letter, say t. So the hacker believes E = $a^{-1}(t-b) \bmod m$. Thus they can try all a,b values that make this true. Another layer to this is trying other common letters like "T" and "A" in case E is not the most frequent in that particular plaintext.

```
Attackers can also assume two letters are most common and solve for a,b
directly since there are two unknowns and two equations. They can try this for
multiple combinations of frequent letters until they find an a,b that decrypts
the whole text (ie. E+A, E+T, T+A)
```