CS3302 Exam

200007413

1

a. Alice sends Bob $3^6 \mod 353 = 23$.

Bob sends Alice $3^{11} \mod 353 = 294$.

Alice calculates the shared key $294^6 \mod 353 = 253$.

Bob calculates the shared key $23^{11} \mod 353 = 253$.

Therefore the shared key is 253.

b. 0x65 (01100101) can be written $f(x) = x^6 + x^5 + x^2 + 1$

With the AES irreducible polynomial $m(x) = x^8 + x^4 + x^3 + x + 1$ we use the extended euclidian algorithm to calculate the multiplicative inverse. We first divide $m(x)$ by $f(x)$:

	x^2+x+1							
$x^6+x^5+x^2+1$	x^8+			x^4+	x^3+		$x+$	1
	x^8+	x^7+		x^4+		x^2		
		x^7+			x^3+	x^2+	$x+$	1
		x^7+	x^6+		x^3+		x	
			x^6+			x^2+		1
			x^6+	x^5+		x^2+		1
				x^5				

Therefore, our quotient is $x^2 + x + 1$, and remainder x^5.

Our multiplicative inverse is then the quotient which is 00000111.

c. The matching formulas would be:

$$x_1 = D(c_1)$$

$$x_i = D(c_i) \oplus c_{i-1} \oplus x_{i-1}$$

This works because calling $D(c_1)$ just get x_1, then calling $D(c_i)$ for all other blocks gets $x_i \oplus c_{i-1} \oplus x_{i-1}$ Therefore, repeating the bitwise XOR with the previous ciphertext block and previous plaintext block will revert the XOR to just get x_i.

d. We cannot decrypt the rest of the message if one block is lost because the next blocks rely on the previous plaintext block which in turn relies on it's previous block till you get to the first block which relies on nothing but the first ciphertext block.

e. Yes we can, because calling XOR doesn't depend on the order it's called, and so the two previous blocks together in any order will suffice to find the next block.

2

a. The fano condition requires no code words are a prefix of another code word. This means that the code words are uniquely decodable.

Therefore, when a set K with the fano condition is complete, then every uniquely decodable code word is in K.

For this to be the case, it must be that every string in B^* is the prefix of a combination of code words. If this were not the case, then if we take out the code words that match in $B*$, we are left with symbols which do not match up to the prefix of any code words, and so not all uniquely decodable combinations of the code alphabet are matched, meaning it is not complete.

If it is safe, then we know that every string of the code alphabet is a mix of codewords, and then symbols which are prefixes to a unique code word, which means that all combinations of the code alphabet are matched meaning K is complete.

QED.

b.

c.

c	a	d	e	b
0.64	0.11	0.1	0.09	0.06
c	eb	a	d	
0.64	0.15	0.11	0.1	
c	ad	eb		
0.64	0.21	0.15		
c	adeb			
0.64	0.36			

c	a	d	e	b
0	100	101	110	111
c	eb	a	d	
0	11	100	101	
c	ad	eb		
0	10	11		
c	adeb			
0	1			

The average length is $(0.64 * 1) + (0.11 * 3) + (0.1 * 3) + (0.09 * 3) + (0.06 * 3) = 0.64 + 0.33 + 0.3 + 0.27 + 0.18 = 1.72$

d.

dog	cat
$\frac{1}{3}$	$\frac{2}{3}$

$$\begin{aligned} H(\tfrac{1}{3}, \tfrac{2}{3}) &= -(\tfrac{1}{3}log_2(\tfrac{1}{3}) + \tfrac{2}{3}log_2(\tfrac{2}{3})) \\ &= -(-1.58496... - 0.58496...) \\ &= 2.16999... \\ &= 2.170 \text{ (3dp)} \end{aligned}$$

$\frac{1}{3} \cdot \frac{1}{5} + \frac{2}{3} \cdot \frac{1}{4} = \frac{3}{20}$ pets are white. Therefore:

$\frac{17}{20}$ pets are black.

$$\begin{aligned} H(\tfrac{3}{20}, \tfrac{17}{20}) &= -(\tfrac{3}{20}log_2(\tfrac{3}{20}) + \tfrac{17}{20}log_2(\tfrac{17}{20})) \\ &= -(-0.4105... - 0.19992...) \\ &= -0.6098... \\ &= -0.610 \text{ (3dp)} \end{aligned}$$

3

a. i.

0	0	0	0	1	1	1
1	0	0	1	1	0	1
0	1	1	1	0	1	1
0	0	1	1	0	0	1

Swap 1st and second row:

1	0	0	1	1	0	1
0	0	0	0	1	1	1
0	1	1	1	0	1	1
0	0	1	1	0	0	1

Swap 2nd and third row

1	0	0	1	1	0	1
0	1	1	1	0	1	1
0	0	0	0	1	1	1
0	0	1	1	0	0	1

Swap third and fourth row

1	0	0	1	1	0	1
0	1	1	1	0	1	1
0	0	1	1	0	0	1
0	0	0	0	1	1	1

Add third to second

1	0	0	1	1	0	1
0	1	0	0	0	1	0
0	0	1	1	0	0	1
0	0	0	0	1	1	1

Add fourth to first

$$\begin{matrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{matrix}$$

QED.

ii. They are equivalent because the matrixes were formed from the basis B, which requires that it is linearly independent.

b. n is 7, as the size of each row is 7bits wide

k is 4 as given by the base B which has 4 elements.

d is 3 because the minimum number of bits to convert one code word to another is 3.

Therefore, L could detect 2 errors, but only correct 1 because if 1 error occured, then it would be closest to the correct value, and if 2 it wouldn't match a code word but you couldn't tell the correct value.

This fits the first parameter of a perfect code, bause $n = 7 = 2d + 1 = 2 * 3 + 1$.

However, is not perfect, because $2^k = 16$ and $\frac{2*7}{1+7} = 1.75$ which is not 16 which is the second requirement for a perfect code.

c. With $m = 110$, we use matrix multiplication with the generator matrix to get: 110101.

Recieving 010101 would require an error on just the first bit and so the probability would be.

$0.015 * (1 - 0.015)^5 = 0.014$ (3dp)

d. H =

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$