QUESTIONS/SOLUTIONS 1999-00

Time allowed - Two Hours.

Candidates should attempt THREE questions.

The intended marks for questions or parts of questions are given in brackets [].

1. (i) Let \( d = 2e + 1 \) be an odd integer. Define \( A_q(n, d) \) to be the maximum value of \( M \) such that there exists a \( q \)-ary \((n, M, d)\) code, i.e. a general code over \( q \) symbols, consisting of vectors of length \( n \) and having \( M \) points. Prove the Hamming bound

\[
A_q(n, d) \sum_{k=0}^{k < e} \binom{n}{k} (q - 1)^k \leq q^n
\]

[10]

(ii) For a channel transmission the 6 bit binary characters \( \{x_1, \ldots, x_6\} \) are sent as 7 bits \( \{x_1, \ldots, x_7\} \) using a lateral parity check bit

\[
x_7 = 1 + \sum_{i=1}^{6} x_i \quad (\text{mod } 2)
\]

Each block of three successive characters is provided with a fourth 7 bit character which acts as longitudinal parity check (the seventh bit of this character being a parity check on the other 6 check bits). With no error correction the probability of a single bit being received in error is \( 10^{-6} \). The bit rate of the channel is \( 10^6 \) bits per sec. Compute the mean time in days between errors when the error correction scheme described is employed. [You may ignore the possibility of errors in the check bits, and assume that more than two errors in a block are so improbable as to be irrelevant.]

[10]

2. (a) Give the decoding algorithm for the Hamming code \( \text{Ham}(r, 2) \). How can the straightforward decoding algorithm be improved? [8]

(b) Consider the binary Hamming code \( C = \text{Ham}(r, 2) \) with \( r = 3 \), and parity check matrix of size \( r \times (2^r - 1) \) whose columns are the distinct non-zero vectors of \( V(r, 2) \) (the vector space of dimension \( r \) over the binary field with elements \{0, 1\}).

(i) Give the parity check matrix of \( C \). [2]

(ii) Rewrite the parity check matrix in standard form. [3]

(iii) Give the generator matrix for \( C \). [3]

(iv) The received word is \( y = \{1, 1, 0, 1, 0, 1, 1\} \). What it is syndrome and how should it be decoded? [4]
3. (a) Describe Shamir’s method for sharing a secret key among \( n \) persons in such a way that any subset of \( t \) or more can recover the key but no subset of less than \( t \) can do so. [5]

(b) In the following case \( n = 4 \) and \( t = 3 \) and Shamir’s method has been used to provide information about the secret key. Persons 1, 2, and 4 decide to recover the key. Their numbers are

\[
f(1) = 7, f(2) = 6, f(4) = 14 \quad \text{and all arithmetic is (mod 19)}.
\]

Find the polynomial and hence the key and also the value of \( f(3) \). [5]

(c) The revolutionary council has 20 members. They decide that their archives can be accessed by any 10 members. To enhance the security of Shamir’s method they decide that the key will not be the constant term of the polynomial, but instead the concatenation of all the coefficients. The government computers can solve \( 10^7 \) systems of linear congruences in 10 variables per second. The council decide that sufficient security is provided if, knowing the prime modulus, it would take the government \( 10^{100} \) years to perform an exhaustive search for the polynomial and hence the key.

(i) How big a prime is required in Shamir’s method if the government attacks by attempting to solve all systems of 10 congruences (mod \( p \)) in 10 variables? [5]

(ii) Using this size prime is it better for the government to try a direct attack by exhaustively searching keys at a rate of \( 10^{10} \) per sec?

[There are approximately \( 3.15 \times 10^7 \) secs per year]. [5]

4 (a) Define Euler's function \( \varphi(n) \) and show that \( \varphi(pq) = (p - 1)(q - 1) \) for any primes \( p \) and \( q \). [4]

(b) Explain the RSA algorithm for public key encryption/decryption. [10]

(c) A public key system which uses the RSA algorithm is based on the modulus \( n = 221 \) and encryption key \( e = 77 \).

(i) What is the decryption key? [3]

(ii) If an encrypted number is 122 what is the original number? [3]
Solutions

1. (i) Prove the Hamming bound

\[ A_q(n, d) \sum_{k=0}^{d} \binom{n}{k} (q-1)^k \leq q^n \]

There are

\[ \binom{n}{d} = \frac{n!}{d!(n-d)!} \]

ways of choosing a vector \( y \) which differs from a given vector \( x \) in exactly \( d \) places.

[2 Marks]

At each of the \( d \) places we can choose the corresponding element of \( y \) to have any one of \( q-1 \) values, so for each vector \( y \) we have \( (q-1)^d \) possibilities.

[2 Marks]

Hence there are

\[ \binom{n}{d} (q-1)^d \]

such vectors in all.

[2 Marks]

If \( C \) is a \( q \)-ary \( (n, M, 2e+1) \) code then any two codewords are at a distance at least \( 2e+1 \) apart and therefore there is no vector at distance \( \leq e \) from both of them.

[2 Marks]

Now the total number of vectors is \( q^n \) and so, counting all vectors at distance \( \leq e \) from each of the \( A_q(n, d) \) codewords, where no codeword is counted twice, we have by the above argument

\[ A_q(n, d) \left( 1 + \binom{n}{1} (q-1) + \binom{n}{2} (q-2)^2 + \ldots + \binom{n}{e} (q-1)^e \right) \leq q^n \]

i.e.

\[ A_q(n, d) \left( \sum_{k=0}^{e} \binom{n}{k} (q-1)^k \right) \leq q^n \]

which proves the Hamming bound.

[2 Marks]

[10 Marks total]

(ii) For a channel transmission the 6 bit binary characters \( \{x_1, \ldots, x_6\} \) are sent as 7 bits \( \{x_1, \ldots, x_7\} \) using a lateral
parity check bit

\[ x_j = 1 + \sum_{i=1}^{6} x_i \pmod{2} \]

Each block of three successive characters is provided with a fourth 7 bit character which acts as longitudinal parity check (the seventh bit of this character being a parity check on the other 6 check bits). With no error correction the probability of a single bit being received in error is \(10^{-6}\). The bit rate of the channel is \(10^6\) bits per sec. **Compute the mean time in days between errors when the error correction scheme described is employed.** [You may ignore the possibility of errors in the check bits, and assume that more than two errors in a block are so improbable as to be irrelevant.] [15]

A single error will be detected and corrected. Double errors can occur in the message bits as shown in the figure.

![Possible locations of double errors.](image)

(a). It might involve two erroneous bits in the same 6 bits of a character. This will not be apparent from the lateral parity check, although two longitudinal check bits will reveal the errors. However, we cannot tell which character has the two errors.

(b). It might be that two of the three characters have an error in the same position. This will not be apparent from the longitudinal parity checks, although two lateral check bits will reveal the errors. In this case we know which two characters are in error but not where the errors occur.

(c). It might be that two errors occur which are not in the same row or column (see figure). In case we still cannot correct the errors since this scenario is also compatible with (d).

Thus two errors occurring *anywhere* in the message bits are uncorrectable using this code.

Thus the probability of a double uncorrectable error in a single block is:

\[ \binom{18}{2} p^2 (1 - p)^{16} = 153 \times 10^{-12} \times (1 - 10^{-6})^{16} \]

A block contains 4 x 7 = 28 bits. So the channel rate of \(10^6\) bits per second corresponds to a block rate of 0.0357 \(\times 10^6\) blocks per sec.

Thus the expected time to an uncorrectable error is:
\[ \frac{1}{0.03757 \times 10^6 \times 153 \times 10^{-12}} = 1.73967 \times 10^5 \text{ sec} \]

= 2.0135 days

[10 Marks Total]
2. (Remark: In terms of the notation for linear codes $n = 2^r - 1$, $k = 2^r - 1 - r$, so $n - k = r$.)

(a) The straightforward decoding algorithm is:

Step 1. When a vector $y$ arrives calculate its syndrome $s = H y \pmod{2}$, where $H$ is the $(n-k) \times n$ parity check matrix.

Step 2. If $s$ is the $(n-k)$-dims zero vector then assume $y$ was the codeword sent.

Step 3. If $s$ is not the $(n-k)$-dims zero vector then, assuming a single error, $s$ gives the binary representation of the error and so the error can be corrected.

[5 Marks]

Improvement. If the parity check matrix $H$ is such that the columns from left-to-right count up in binary then this error correction algorithm becomes even simpler: we just look to see which column of $H$ is the syndrome, if it is the $j$th column then (because the columns of $H$ are counting up in binary) the error is in the $j$th bit of the received word.

[3 Marks]

[8 Marks Total]

(b) (i) The parity check matrix is (by definition):

\[
\begin{bmatrix}
0, 0, 0, 1, 1, 1, 1 \\
0, 1, 1, 0, 0, 1, 1 \\
1, 0, 1, 0, 1, 0, 1
\end{bmatrix}
\]

[2 Marks]

(ii) The parity check matrix $H = [B, I_{n-k}]$ in standard form (permute columns to get identity matrix as right block) is:

\[
\begin{bmatrix}
0, 1, 1, 1, 1, 0, 0 \\
1, 0, 1, 1, 0, 1, 0 \\
1, 1, 0, 1, 0, 1, 1
\end{bmatrix}
\]

[3 Marks]

(iii) The generator matrix $G = [I_k, -B^T]$ is:

\[
\begin{bmatrix}
1, 0, 0, 0, 0, 1, 1 \\
0, 1, 0, 0, 1, 0, 1 \\
0, 0, 1, 0, 1, 1, 0 \\
0, 0, 0, 1, 1, 1, 1
\end{bmatrix}
\]

[3 Marks]

(iv) The received word is $y = \{1, 1, 0, 1, 0, 1, 1\}$. The syndrome is

\[ s = Hy^T = \{1, 1, 0\} \]

[2 Marks]

This indicates an error in position 6. Thus we should flip the $6$th bit in $y$ in this position to get:
\[1, 1, 0, 1, 0, 0, 1\]

[2 Marks]

[4 Marks Total]

[Check if this new \( y \) really is a codeword. Syndrome is

\[ s = H \{1, 1, 0, 1, 0, 0, 1\}^T = \{0, 0, 0\} \]

which is correct.]
3. (a) Shamir’s method. The trusted administrator chooses a polynomial of degree \( t - 1 \) with integer coefficients \((\text{mod } p)\), where \( p \) is a large prime \( p >> n \). The constant term in this polynomial is \( K \), the secret number, and the other coefficients are chosen at random \((\text{mod } p)\).

Let this polynomial be
\[
f(x) = K + a_1 x + a_2 x^2 + \ldots + a_{t-1} x^{t-1}
\]

The \( n \) recipients are allocated the identifying labels 1, 2, ..., \( n \) and the \( ith \) recipient is told the value of \( f(i) \) \((\text{mod } p)\). In order to recover the secret number \( K \) it is sufficient to find the polynomial \( f(x) \), because then \( K = f(0) \).

- This can be done if and only if \( t \) pairs of values \((i, f(i))\) are available.

(b) From the question
\[
\begin{align*}
f(1) &= K + a_1 + a_2 \equiv 7 \pmod{19} \\
f(2) &= K + 2a_1 + 4a_2 \equiv 6 \pmod{19} \\
f(4) &= K + 4a_1 + 16a_2 \equiv 14 \pmod{19}
\end{align*}
\]

Solving these congruences by Gaussian elimination gives \( \{K, a_1, a_2\} = \{5, 13, 8\} \). Therefore secret key is \( K = 5 \) and \( f(3) = 2 \).

[5 Marks]

(c) (i) To be effective this method must use a sufficiently large prime to prevent an exhaustive check of all possible solutions. If there are \( n \) people and any group of \( t \) can recover the key then any person could have one of \( p-1 \) numbers so we should have to check \((p-1)^t\) \( t \)-tuples, where we suppose these are the numbers for persons 1 to \( t \). (If we don’t suppose this then the computation increases by a factor of \( ^tC_n \) ) For each \( t \)-tuple we have to solve a system of linear congruences \((\text{mod } p)\). We are told that knowing \( p \) the government computers can solve \( 10^9 \) such systems of congruences per sec (unlikely), and the revolutionary council have resolved that they require that it should take the government \( 10^{100} \) years to exhaustively search. There are approximately \( 3.15 \times 10^7 \) secs per year (given). Hence the requirement is for a number of systems of congruences around
\[
3.15 \times 10^7 \times 10^9 \times 10^{100} = 3.15 \times 10^{116}
\]

Thus \((p-1)^t\) should be at least as large as this. A quorum of the revolutionary council is defined to have size \( t = 10 \). Then we should have \((p-1)^{10}\) approx \( 10^{116} \). i.e. \( p \) about \( 10^{11} \).

[5 Marks]

(ii) With \( p \) about \( 10^{11} \) the concatenation of all the 10 coefficients could be as large as \( 10^{10} \) and so a direct search for the key, checking \( 10^{10} \) keys per sec, would take approximately
\[
10^{10} \times (3.15 \times 10^7 \times 10^{10}) = 2.8 \times 10^{100} \times 10^{10} \times 10^{10} = 2.8 \times 10^{90} \text{ years}
\]

i.e it is more efficient for the government to directly search keys than to try to find the polynomial.

[5 Marks]
4. (a) \( \varphi(n) \) is the number of integers \( k \) (\( 1 \leq k \leq n \)) such that \( k \) and \( n \) are coprime (have no non-trivial factors in common).

[2 Marks]

For a prime \( p \) the numbers between 1 and \( p \) that are coprime to \( p \) are 1, 2, ..., \( p-1 \). Hence \( \varphi(p) = p - 1 \). Similarly the numbers between 1 and \( pq \) that are coprime to \( pq \) are

\[
1, \ldots, p-1; 2.1, 2.2, \ldots, 2(p-1); 3.1, 3.2, \ldots, 3(p-1); \ldots,(q-1).1 \ldots(q-1)(p-1)
\]

so that \( \varphi(pq) = (p - 1)(q - 1) \).

[2 Marks]

[4 Marks Total]

[Multiplicative properties of \( \varphi \) were not discussed.]

(b) The RSA public key encryption system

The system is based on choosing three integers \((e, p, q)\) where \( p \) and \( q \) are large primes \((= 10^{200})\) and \( e \) is coprime to \((p - 1)(q - 1)\).

Let \( n = pq \). Then \( e \) and \( n \) are made public and only the owner of the system knows \( p \) and \( q \).

[2 Marks]

To encrypt the message:

1. Convert the message into a stream of decimal digits (the method for doing this is not secret, indeed it must be known).
2. Cut the stream into blocks of equal length and convert these to decimal numbers \( m_1, m_2, \ldots \), where \( m_i < n \).
3. Raise each number \( m_i \) to the power \( e \) and replace \( m_i^e \) by its remainder \( r_i \pmod{n} \) (so \( m_i^e \equiv r_i \pmod{n} \) in mathematical notation).
4. The encrypted message is then the stream \( r_1 r_2 r_3 \ldots \)

[4 Marks]

To decrypt the message:

1. Let \( d \) be the number that has the property

\[
ed \equiv 1 \pmod{(p - 1)(q - 1)}
\]

(this is not too hard to compute, it can be done by using Euclid's algorithm, and can be worked out once and for all and saved).
2. Then

\[
r_i^d \equiv (m_i^e)^d \equiv m_i^{ed} \equiv m_i^{\varphi(n) - 1} \equiv m_i \pmod{n}
\]

by the Euler-Fermat theorem, and the decrypted message is \( m_1 m_2 m_3 \ldots \)

[4 Marks]
(c) Factorising 221 gives \( n = 13 \times 17 = 221 \). So \( p = 13 \), \( q = 17 \). Hence \( \varphi(pq) = 12 \times 16 = 192 \). The public key exponent is \( e = 77 \). Hence the decryption key \( d \) is the solution to \( 77d \equiv 1 \pmod{192} \).

(i) A few trials (or a more systematic attack) quickly gives \( d = 5 \).

[3 Marks]

(ii) The encrypted word is 122 so the decrypt is \( 122^5 \equiv 5 \pmod{221} \).

[3 Marks]