Suggested solutions to final exam in CRYPTOGRAPHY on 14 December 2005.

Problem 1

a) An encryption function must be bijective (invertible) for all fixed keys K. Clearly, $E_K(M) = M$ and $E_K(M) = M + K$ are bijective. The others are not bijective for all K.

b) Unicity distance $N_0 = H(K)/D$

$$\begin{aligned} H(K) &= \log 10! \\ D &= H_0 - H_\infty = H_0 - H(M) \\ H_0 &= \log 10 \\ H(M) &= -2\tfrac{1}{4}\log\tfrac{1}{4} - 8\tfrac{1}{16}\log\left(\tfrac{1}{16}\right) \\ &= 3 \\ D &= \log 10 - 3 \end{aligned}$$

So, $N_0 = H(K)/D = (\log 10!)/(\log 10 - 3) \approx 68$.

Problem 2

a) $S(D) = \frac{2+D^2}{1-D^4}$ Calculating $\gcd(2+D^2, 1-D^4) = \ldots$ gives $S(D) = \frac{2}{1+D^2}$ and since nominator and denominator are relatively prime, the shortest LFSR is $C(D) = 1 + D^2$ of length 2.

b) Since $p(x)$ is irreducible but not primitive we cannot use α as generator. Instead, use $\alpha + 1$ as generator for the multiplicative group, i.e., $(\alpha+1), (\alpha+1)^2 = 2\alpha, (\alpha+1)^3 = 2\alpha + 1, \ldots$, etc.

The Berlekamp-Massey algorithm then gives $C(D) = 1 - D - (2\alpha+2)D^2 - 2D^3$ and $L = 3$ after some work.

Problem 3

a) The reconstruction can be made by using the formula given in b), i.e.,

$$K = 4\frac{2\cdot 3}{(2-1)(3-1)} + 5\frac{1\cdot 3}{(1-2)(3-2)} + 6\frac{1\cdot 2}{(2-3)(1-3)} = 4\frac{2\cdot 3}{2} + 5\frac{1\cdot 3}{-1} + 6\frac{1\cdot 2}{2} = 4\cdot 3 - 5\cdot 3 + 6 = 3.$$

b) Assume that k participants (wlog numbered 1 to k) should reconstruct k. They each have access to a point (x_i, y_i), where $y_i = a(x_i)$, x_i is a public value, and $a(x)$ is an unknown polynomial of degree at most $k-1$. Now we need to prove that given the k different points (x_i, y_i), the polynomial $a(x)$ is given by the expression

$$A(x) = \sum_{i=1}^{k} y_i \prod_{1\le j\le k, j\ne i} \frac{x - x_i}{x_i - x_j}.$$

We evaluate the right hand side in point x_l, giving

$$A(x_l) = \sum_{i=1}^{k} y_i \prod_{1\le j\le k, j\ne i} \frac{x_l - x_j}{x_i - x_j}.$$

In the inner product, j runs through all values except i, meaning that the product is 0 for all i, $i \ne l$. So

$$A(x_l) = y_l \prod_{1\le j\le k, j\ne i} \frac{x_l - x_j}{x_l - x_j} = y_l.$$

This proves that $A(x_l) = a(x_l)$ for $l = 1..k$. Since a polynomial of degree at most $k-1$ is uniquely defined by k points on the polynomial (we do not require you to prove this), we have $A(x) = a(x)$. Since the secret K is the constant term a_0 in the polynomial and $a_0 = a(0)$ we get the desired formula.

Problem 4

a) The decryption exponent is defined by $e \cdot d = 1 \pmod{\phi)(n)}$. Here $\phi(n) = 24810036$ and $d = e^{-1}$ in $\mathbb{Z}_{\phi(n)}$. Use Euclidean algorithm etc to compute $d = 19848029$.

b) If you are clever, you see that $10000 = 10^5$, so $M = 10$. The standard way is otherwise to compute $M = C^d = 100000^{19848029} \bmod 24820049 = 10$.

c) If you want to generate a digital signature to a message M, you use the secret RSA exponent d and compute $S = M^d$ as your signature. When someone receives a signed message (M, S) the correctness of the signature is checked by computing S^e and comparing with M.

d) The above description is not really secure without some further modification. The most common one is the use of a hash function. A hash function h is a deterministic (without key) function taking an arbitrarily long message and hashes it to a fixed length result called the message digest. So given a message M, we first hash it to $h(M)$, then we sign this using RSA, computing $S = h(M)^d$. On the receiver side, a signed message (M, S) is checked by first computing $h(M)$, then computing S^e and comparing with $h(M)$.

If we can find two different messages M, M' hashing to the same value $(h(M) = h(M'))$ then one can ask someone to sign a message M and later replace M with M'. The signature will be valid on both. A collision free hash functions means that it is computationally impossible to find two messages hashing by h to the same value.

Problem 5

a) Start by factoring the polynomial to $(1+2x)^2(1+2x+x^3)$. Next, compute the period of $1+2x$ and $(1+2x+x^3)$. The period of $1+2x$ is $T_1 = 1$ and the period of $(1+2x)^2$ is $T_2 = 3$. This gives a cycle set $1(1) \oplus 2(1) \oplus 2(3)$ for the first part.

For the irreducible $p(x) = (1+2x+x^3)$ we see that if $p(\alpha) = 0$ then $\alpha^2 \neq 1$ as well as $\alpha^{13} \neq 1$. Since $ord(\alpha)|(3^3 - 1)$ we must have $ord(\alpha) = 26$. When the polynomial is irreducible, the order of α and the period of the polynomial are the same. This gives in the end

$$3(1) \oplus 3(26) \oplus 2(3) \oplus 2(78).$$

b) In a stream cipher the ciphertext is generated as $C_i = M_i + Z_i$, $i \geq 0$. In this case the sequence Z_i, $i \geq 0$ is a sequence directly from an LFSR with connection polynomial $p(x)$. If we let s_0, s_1, s_2, s_3, s_4 denote the initial state, then the output of the LFSR is

$$z_0 = s_0, z_1 = s_1, z_2 = s_2, z_3 = s_3, z_4 = s_4, z_5 = 2s_0 + 2s_1, z_6 = 2s_1 + 2s_2, z_7 = 2s_2 + 2s_3,$$

$$z_8 = 2s_3 + 2s_4, , z_9 = 2s_4 + 2s_0 + 2s_1, z_{10} = 2s_0 + s_1 + 2s_2, \ldots$$

As $M_i = 0$, $i = 0, 2, 4, 6, 8, 10$, we have $C_i = Z_i$ $i = 0, 2, 4, 6, 8, 10$. This gives $z_0 = s_0 = 0, z_2 = s_2 = 0, z_4 = s_4 = 0, z_6 = 2s_1 + 2s_2 = 2, z_8 = 2s_3 + 2s_4 = 2, z_{10} = 2s_0 + s_1 + 2s_2 = 2$ or

$$(s_0, s_1, s_2, s_3, s_4) = (0, 1, 0, 1, 0).$$
