Short solutions to final exam in CRYPTOGRAPHY on 15 December 2010.

Problem 1

a) $E_K(M) = M$ and $E_K(M) = M + K$, because they are the only bijective mappings (unique decoding) for any fixed key K.

(5p)

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

$$\begin{aligned} H(K) &= \log_2(10) \\ D &= H_0 - H_\infty = H_0 - H(M) \\ H_0 &= \log_2(10) \\ H(M) &= 6 \cdot \tfrac{1}{8}\log_2(8) + 4 \cdot \tfrac{1}{16}\log_2(16) = 13/4. \\ D &= \log_2(10) - 13/4 \approx 0.0719 \end{aligned}$$

So, the unicity distance is $N_0 = \frac{H(K)}{D} = \frac{\log_2(10)}{\log_2(10)-13/4} \approx 46$.

(5p)

Problem 2

a) A Shamir scheme (3,7) over $\mathbf{F}_{101}$, the secret key $K = a_0$ and $x_i = i$. We use $x_1 = 1, x_3 = 3, x_7 = 7$ and shares $y_1 = 1, y_3 = 10, y_7 = 100$.

$$\begin{pmatrix} 1 \\ 10 \\ 100 \end{pmatrix} = \begin{pmatrix} a_0 + a_1 \cdot 1 + a_2 \cdot 1^2 \\ a_0 + a_1 \cdot 3 + a_2 \cdot 3^2 \\ a_0 + a_1 \cdot 7 + a_2 \cdot 7^2 \end{pmatrix},$$

Gaussian elimination gives the solution to a_0, a_1, a_2, which will be

$$mK = a_0 = 8^{-1} \cdot 44 = 56 \bmod 101,$$

where 8^{-1} was computed using Euclidean algorithm.

(5p)

b) [M is a 3-tuple, $M = (s_1, s_2, t)$, *corrected at exam*]
$P(M'$ accepted $|M$ observed $)=$

$$= \frac{\left(\text{Number of keys } (k_1,k_2) \text{ for which } \left\{ \begin{array}{l} t = k_1 + s_1 k_2 + s_2 k_2^2 \\ t' = k_1 + s_1' k_2 + s_2' k_2^2 \end{array} \right. \right)}{\text{Number of keys } (k_1,k_2) \text{ for which } t = k_1 + s_1 k_2 + s_2 k_2^2} =$$

$$= \frac{\left| \left\{ (k_1,k_2) : \left\{ \begin{array}{l} t = k_1 + m_1 k_2 + m_2 k_2^2 \\ t' - t = (m_1' - m_1)k_2 + (m_2' - m_2)k_2^2 \end{array} \right. \right\} \right|}{3} =$$

$$\leq \tfrac{2}{3}.$$

$P_S = \max_{M,M',M\neq M'} P(M' \text{ accepted}|M \text{ observed}) = \frac{2}{3}$, by finding one pair for which equality holds.

(5p)

Problem 3

b) B-M algorithm gives $C(D) = 1 + D + D^2 + D^6$.

(5p)

b) Continue using the B-M algorithm, or set up the quotient

$$S(D) = \frac{s_0 + s_1 D + \ldots s_{T-1} D^{T-1}}{1 - D^T}.$$

(In this case it happens that B-M algorithm gives much less work.

The solution is $C(D) = 1 + D + \alpha D^2$. (5p)

Problem 4

a) We know that $p(x)$ should be of degree 3 to build $\mathbb{F}_{11^3}$. Irreducible means thaty it cannot be factored. If $p(x)$ is not irreducible then one of the polynomial factors must have degree one, i.e. $p(\chi) = 0$ for some $\chi \in \mathbb{F}_{11}$. Choose a random polynomial and test it, until it has no zero point.

For example $p(x) = x^3 + x^2 + x + 2$ is irreducible, because $p(0) = 2, p(1) = 5, p(2) = 5, \ldots$ (5p)

b) You need to construct a deBruijn sequence (project) with symbols in $\mathbb{F}_{11}$. The difficult part is to find a primitive polynomial [noone succeeded on exam]. With an irreducible polynomial $p(x)$, we need to test if it is primitive (for $p(\alpha) = 0$, α is a primitive element). We know that $\text{ord}(\alpha)|(11^3 - 1)$ and $11^3 - 1 = 2 \cdot 5 \cdot 7 \cdot 19$. Note that if $\alpha^{1330/2} \neq 1, \alpha^{1330/5} \neq 1, \alpha^{1330/7} \neq 1, \alpha^{1330/19} \neq 1$ then α must be a primitive element. Use square and multiply, or compute $\beta_1 = \alpha^2, \beta_2 = \beta_1^5, \beta_3 = \beta_1^7$ to get $\beta_3 = \alpha^{1330/19}$, etc.

For example $p(x) = x^3 + x^2 + x + 3$ is primitive in $\mathbb{F}_{11}$.

Finally, we need to modify the feedback to include the all zero state and draw a picture of this, specifying the details. This is the same as in the project.

(5p)

Problem 5

a)

$$d = e^{-1} \bmod \phi(n).$$

Euclidean algorithm gives $d = 19848029$.

(3p)

b) $M = 10$ (the observant can see that $C = 10^5$ and $e = 5$. Otherwise square-and-multiply).

(3p)

c) The signature S is generated by "encrypting" using the secret key d, i.e.,

$$S = M^d \bmod n.$$

Then (M, S) is transmitted and receiver verifies that $S^e = M \bmod n$.

(2p)

d) Use of a hash function allows variable-sized messages to be hashed. It also protects against some attacks on the basic version in c).

The problem of finding two messages mapping to the same hash value is difficult (complexity around $2^{l/2}$ if l is the bit length of the hash value) for a collision-free hash function.

(2p)
