MATC16 Cryptography and Coding Theory
Gábor Pete
University of Toronto Scarborough

Final exam solutions

Problem 1. (2+2+2 points)

(a) What is the size of the keyspace for the affine cipher over the English alphabet?

$x \mapsto \alpha x + \beta \pmod{26}$, where $\gcd(\alpha, 26) = 1$ must be for invertibility. Hence 12 choices for α and 26 for β, that's altogether $12 \cdot 26 = 312$.

(b) Show that the composition of two affine ciphers is again an affine cipher.

If the first key is (α_1, β_1), the second is (α_2, β_2), then the composition is

$$x \mapsto \alpha_1 x + \beta_1 \mapsto \alpha_2(\alpha_1 x + \beta_1) + \beta_2 = \alpha_1\alpha_2 x + (\alpha_2\beta_1 + \beta_2)\,.$$

(c) What do you think the largest problem is with the security of the Hill cipher?

The problem is that a known plaintext attack gives away the key easily, with not much plaintext. Indeed, in an $n \times n$ Hill cipher with encryption key matrix K, given n^2 pairs of known plaintext-ciphertext letters such that the resulting $n \times n$ plaintext matrix P is invertible, the matrix equation $PK = C$ is solved by $K = P^{-1}C$. The suitable n^2 plaintext letters will probably be found in a not much longer piece of plaintext.

By the way, the system has diffusion, hence frequency attack does not work. Whether it has confusion, is more subtle. It has reasonable confusion in the sense that each ciphertext letter depends on a large part of the key (this is also what the book says). However, this dependence is not "as complex as possible", in the sense that it is in fact linear, and this linearity is what makes the above known plaintext attack possible.

Problem 2. (2+2 points)

(a) About how many times more time does a brute force key search take against a 112-bit DES than against a 56-bit DES?

The key space sizes are 2^{112} versus 2^{56}, respectively. Hence the ratio of the time amounts that the brute force searches take is about $2^{112}/2^{56} = 2^{56} \approx 7 \cdot 10^{16}$.

(b) What is the Double version of the 56-bit DES, and why is it much less secure than a single 112-bit DES? (Enough to sketch your attack against Double DES, but not enough to give just the name of it.)

Double DES: the plaintext m is encoded twice, with different keys: $m \mapsto E_{k_2}\big(E_{k_1}(m)\big)$.

This composition is NOT encryption by a single DES key ("DES is not a group"), hence the key-space size of Double 56-DES is basically 2^{112}, the same as of the single 112-DES.

However , Double DES can be attacked by "meet-in-the-middle", which takes not much more time than 2^{56} (but it requires a lot of memory in turn).

This attack: Given a plaintext-ciphertext pair (m, c), with $c = E_{k_2}\big(E_{k_1}(m)\big)$ with some unknown keys k_i, write two lists of length 2^{56} each:

$$\big\{E_k(m) : k \in \{0,1\}^{56}\big\} \text{ and } \big\{D_k(c) : k \in \{0,1\}^{56}\big\}.$$

There will certainly be a common element between these lists: $E_{k_1}(m) = D_{k_2}(c)$.

However, there is the question of how many common elements there are (which you weren't really expected to discuss, at least not in this detail). If m and c were not long, say at most 56 bits, then these lists would be longer than the number of all possible different strings on them, hence it would be quite likely that both lists would contain *all* possible strings, hence we wouldn't have learned anything! But if the length of m and c are $56 + k$ bits, then each list has about 2^{-k} proportion of all possible strings, and the expected number of common strings is $2^{56+k}2^{-2k} = 2^{56-k}$. (Here I'm assuming that the DES outputs are very random-looking.) Well, we know that 56-bit DES operates on blocks of length 64, hence $k = 8$, which means that we have **about 2^{48} common elements** on the lists! That sounds huge, but the remedy is easy: we have just got reduced by a factor of 2^8 the set of all 2^{56} possible keys, hence if we repeat the procedure a bit more than 7 times, with new (m, c) pairs, then by the end only the true (k_1, k_2) key pair will remain.

Problem 3. (3 points)

Find the linear recursion defining the sequence 0101110 0101110 ... of period 7.

If a recursion has order k, so that $x_{n+k} \equiv c_0x_n + c_1x_{n+1} + \cdots + c_{k-1}x_{n+k-1} \pmod 2$, then the resulting sequence will have at most period $2^k - 1$. Hence the period 7 means that $k \geq 3$, and we do not have to check if an order 2 recursion works. For order 3:

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix},$$

which gives the solution $c_0 = c_1 = 1$, $c_2 = 0$, hence $x_{n+2} = x_n + x_{n+1}$.

Note that the trivial recursion $x_{n+7} = x_n$ is not a proper solution to the problem. In real applications, where you start, say, with a very simple order 31 recursion, and get a sequence of period $2^{31} - 1$, you want to find the order 31 recursion, not the order $2^{31} - 1$ one, because in the first case you will need to know 31 consecutive elements of the sequence to generate the entire sequence, while in the second case you would need $2^{31} - 1$ elements.

Problem 4. (3 points)

What is the multiplicative inverse of x^{15} in $\mathbb{Z}_2[x] \pmod{x^4 + x + 1}$?

You start the Euclidean algorithm for polynomials: $x^{15} : (x^4 + x + 1) = x^{11} + x^8 + x^7 + x^5 + x^3 + x^2 + x + 1$, with remainder 1. This means that $x^{15} \equiv 1 \pmod{x^4 + x + 1}$, hence its inverse is 1.

Problem 5. (3+3 points)

(a) Alice and Bob have the same modulus n for RSA, and encryption exponents e_A and e_B with $\gcd(e_A, e_B) = 1$. Charles sends them the same message m encrypted with these keys, resulting in the ciphertexts c_A and c_B. Eve intercepts both c_A and c_B. How can she find m?

The ciphertexts are given by $c_A \equiv m^{e_A}$ and $c_B \equiv m^{e_B} \pmod{n}$. By $\gcd(e_A, e_B) = 1$, Eve can find integers s, t with $e_A t + e_B s = 1$. Thus, she can compute $c_A^t c_B^s \equiv m^{e_A t + e_B s} = m \pmod{n}$.

(b) How do you achieve authentication and non-repudiation in RSA?

When Alice wants to send m to Bob, instead of just sending $m^{e_B} \pmod{n_B}$ using his public key, as in simple RSA, she first computes $m' \equiv m^{d_A} \pmod{n_A}$ with her private key, then sends $(m')^{e_B} \pmod{n_B}$ to Bob. This Bob can decrypt with his private key d_B, thus get m', and from that can get m using Alice's public key e_A. Since d_A is known only to Alice, if the resulting m makes sense, that's a proof for Bob and for the world that only Alice could have sent the message, hence we have authentication and non-repudiation.

Problem 6. (3 points) Construct a hash function using the Cipher Block Chaining mode of operation of DES.

Let's say for concreteness that our DES has a 56-bit key K, operating on blocks of 64 bits, although that nowadays would not give a secure enough hash. Let's chop our plaintext P (that we want to hash) into blocks of 64 bits, $P_1, P_2, \ldots, P_m$. Set a block C_0 of 64 bits in some arbitrary way, say, all zeros. Then let $C_i := E_K(C_{i-1} \oplus P_i)$ for $i = 1, 2, \ldots, m$, recursively, where E_K is the DES encryption function with the key K, and $\oplus$ is bitwise addition (mod 2). Finally, $h(P) := C_m$ is the output of the hash. (The CBC mode of operation does the same, except that there the resulting ciphertext is all of $C_1, C_2, \ldots, C_m$.)

(This is fast to compute because DES is fast to compute. It's one-way and strongly collision free because of the good cryptographic properties of DES, namely that it resists even a known plaintext attack. Here we ignore the fact that the 56-bit key is actually too short nowadays.)

Problem 7. (4 points)

Test the primality of $n = 881$ with a Miller-Rabin test.

$881 - 1 = 880 = 2^4 55$. Let's pick a random base, say 107. (Note again that 2 is not a uniform random number between 2 and 880, neither 3 is.) By writing 55 in base 2 and repeated squaring, it is easy to get $107^{55} \equiv 387 \pmod{881}$. This is not ± 1, so need to continue. $387^2 \equiv 880 \equiv -1 \pmod{881}$, so the test says 881 is probably a prime.

Problem 8. (5 points)

Factor $n = 2773$ using the elliptic curve $y^2 \equiv x^3 + 4x + 4 \pmod{n}$ and the point $P = (1, 3)$ on it. Here are the formulas for point addition on a curve $y^2 = x^3 + bx + c$: if $P_3 = P_1 + P_2$ and $P_i = (x_i, y_i)$ for $i = 1, 2, 3$, then

$$x_3 = m^2 - x_1 - x_2, \qquad y_3 = m(x_1 - x_3) - y_1,$$

where $m = (y_2 - y_1)/(x_2 - x_1)$ if $P_1 \neq P_2$, while $m = (3x_1^2 + b)/(2y_1)$ if $P_1 = P_2$.

We first compute $P+P$. By the second formula for the slope m above, $m \equiv 7/6 \equiv 7 \cdot 2311 \equiv 2312 \pmod{2773}$. Then get $P + P = (1771, 705)$.

Then compute $3P = 2P + P$. By the first formula for the slope, $m \equiv 702/1770 \pmod{2773}$ should be. But the Euclidean algorithm gives gcd(1770,2773)=59. Hence we get the factorization $2773 = 59 \cdot 47$.

Problem 9. (2+2+2 points)

(a) What is the difference between a key agreement and a key distribution protocol?

In a key agreement protocol the participants establish the key together during the protocol, neither having a decisive role. In key distribution, one of the participants (or the key distribution centre) generates the key alone, then sends it (securely) to all the partners.

(b) Describe the Diffie-Hellman key agreement protocol.

Alice and Bob choose a large random prime p and primitive root $\alpha \pmod{p}$. These are public. Privately, Alice chooses integer x and Bob y. Then Alice sends $\alpha^x \pmod{p}$ to Bob, and Bob sends $\alpha^y \pmod{p}$ to Alice. They both can compute now the joint key $(\alpha^x)^y = (\alpha^y)^x \pmod{p}$.

(c) Describe the intruder-in-the-middle attack against Diffie-Hellman.

Eve intercepts both α^x and $\alpha^y \pmod{p}$ when sent. She replaces both of them with $\alpha^z \pmod{p}$, with her own z. Alice and Bob receive this, without knowing they are not receiving what they were supposed to. So they will use $(\alpha^z)^x$ and $(\alpha^z)^y \pmod{p}$ as keys, respectively. Since Alice has α^x and α^y, she can compute the keys $(\alpha^z)^x$ and $(\alpha^z)^y \pmod{p}$. Hence, by intercepting the messages encoded with these keys, she can decrypt them, read them, then encode them with the other key and send them to the original addressee. This way she can read everything without Alice or Bob noticing her existence, and can also send fake messages. Note that Alice and Bob don't have the same keys, hence they cannot communicate now without Eve being in between.