MATC16 Cryptography and Coding Theory
Gábor Pete
University of Toronto Scarborough
gpete at utsc dot utoronto dot ca

Solutions for HW 2

Problem 1. A common way to store passwords on a computer is to use DES with the password as the key to encrypt a fixed plaintext (usually $00\cdots 0$). The ciphertext is then stored in the file. When you log in, the procedure is repeated, and the ciphertexts are compared. Why is this method more secure than the similar-sounding method of using the password as the plaintext and using a fixed known key (for example, $00\cdots 0$)? **(1 pt)**

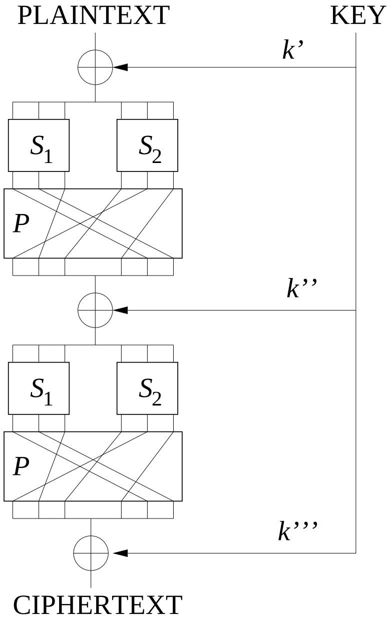
Solution. With the first method, if one steals the file of encrypted passwords, plus knows the fixed plaintext (e.g., by reading the documentation of the password software), then has a plaintext-ciphertext pair for each key, but won't be able to find the key, because DES is safe against such an attack. With the second method, by learning the fixed key from somewhere, one could simply decrypt any ciphertext (since DES is a symmetric key system), thus stealing the encrypted password file would give away the passwords.

Problem 2. Viewing the affine cipher as a double encryption, first multiplication by α, then shift by β, describe how a meet-in-the middle attack on a known plaintext-ciphertext pair works. Is this faster here than brute force key search? **(2 pts)**

Solution. Given plaintext p and ciphertext c, we have $\alpha p + \beta \equiv c \pmod{26}$. Here p and c represent long texts, and $\alpha p+\beta$ is understood letter by letter. (If you had just a pair of single letters, then there are several possible (α, β) keys that work, so you couldn't find the key.) α has 12 possible values, β has 26. Make the lists $L_1 = \{\alpha p : 0 \le \alpha < 26, \gcd(\alpha, 26) = 1\}$ and $L_2 = \{c - \beta : 0 \le \beta < 26\}$. There must be a joint element in the two lists, and if the texts are long enough so that they determine the key, then there can be only one joint element, given by some specific α and β. That pair is the key.
We had to compute $12 + 26$ texts, while brute force search would have needed computing $12 \cdot 26$ texts, so, ignoring the time to find the common element in the list, meet-in-the-middle is faster. (Note that the lengths of the texts needed in the two methods to determine the key uniquely are the same, say, t (around 2 or 3 or 4 or something like that), so in some sense it's $(12 + 26)t$ versus $12 \cdot 26 \cdot t$ computations, but that gives the same result.)

Problem 3. Consider the Cipher Block Chaining (CBC) mode of operation for some block cipher (say, AES), applied to the plaintext P with blocks $P_1, P_2, \ldots, P_n$. If an error occurs in the transmission of a ciphertext block C_j from Alice to Bob, but all other blocks are transmitted correctly, how many blocks will be affected at decryption? **(2 pts)**

Solution. Since $C_{j+1} = E_K(C_j \oplus P_{j+1})$ is the encryption rule, Bob decrypts by $P_{j+1} = D_K(C_{j+1}) \oplus C_j$. If C_j is wrong, then P_j and P_{j+1} will be messed up in the decryption, nothing else.



Problem 4. Consider the substitution-permutation network depicted on the right, encrypting 6-bit plaintexts. The P-box is shown on the picture; the S-boxes act by multiplying row vectors from the right by the following matrices:

$$S_1 = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \text{ and } S_2 = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}.$$

A 6-bit key $\underline{k} = k_1 k_2 \ldots k_6$ gives the three rounds keys $\underline{k}' = k_1 k_3 k_5 k_2 k_4 k_6$, $\underline{k}'' = k_5 k_6 k_3 k_4 k_1 k_2$, and $\underline{k}''' = k_6 k_1 k_4 k_3 k_2 k_5$.

Choose a pair of random 6-bit sequences, $\underline{x}$ and $\underline{y}$; say, flip coins or take your student ID (mod 64) and rewrite the result in binary. Assume that the plaintext $\underline{x}$ gets encrypted into the ciphertext $\underline{y}$. Find the key! (Hint: each transformation here is linear, acting on vectors of length 6.) **(4 pts)**

Solution. In a SP network everything is linear, except possibly the S-boxes, but now those are also linear, so this whole cryptosystem $\underline{y} = E_{\underline{k}}(\underline{x})$ is just a linear system of six equations, with six unknowns: the coordinates of $\underline{k}$. We will write these equations elegantly with matrices (which makes the argument easier to follow, but is not necessary for full credit.) We can combine S_1 and S_2 in one matrix, and write P as a permutation matrix:

$$S = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}, \qquad P = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

Can also write the round key permutations as matrices: $\underline{k}' = \underline{k}K'$, etc, with

$$K' = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad K'' = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad K''' = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Then,

$$\begin{aligned} \underline{y} &= \big((\underline{x} + \underline{k}K')SP + \underline{k}K''\big)SP + \underline{k}K''' \\ &= \underline{x}SPSP + \underline{k}K'SPSP + \underline{k}K''SP + \underline{k}K''' , \end{aligned}$$

where $+$ is coordinate-wise addition (mod 2), and

$$SPSP = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}, \qquad K'SPSP + K''SP + K''' = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

Then,

$$\underline{k} = (\underline{y} + \underline{x}SPSP)\left(K'SPSP + K''SP + K'''\right)^{-1},$$

where the inverse of that matrix does exist:

$$(K'SPSP + K''SP + K''')^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

So, for any $\underline{x}, \underline{y}$ one can get the unique key $\underline{k}$, let me not do an example.

Problem 5.

(a) Show that if p is a prime and $1 \le k \le p-1$, then $p \,\Big|\, \binom{p}{k} = \frac{p!}{k!\,(p-k)!}$ **(1 pt)**

(b) Using part (a), show that if p is a prime, then $x^p + 1$ is a reducible polynomial in $\mathbb{Z}_p[x]$. (Hint: consider first the $p = 2$ case.) **(1 pt)**

(c) How many elements does $\mathbb{Z}_3[x]$ (mod $x^3 + 1$) have? Show that with the usual $+$ and $\cdot$ operations it is not a field. **(2 pts)**

(d) Show that the polynomial $x^2 + 1$ is irreducible in $\mathbb{Z}_3[x]$. **(1 pts)**

(e) Take your student ID, $a_8 a_7 \dots a_0$. What is the polynomial $a_8 x^8 + a_7 x^7 + \cdots + a_1 x + a_0$ in the finite field $\mathbb{Z}_3[x]/(x^2 + 1)$? What is its multiplicative inverse? **(3 pts)**

Solution. (a) In the denominator of $\binom{p}{k}$ we see only factors between 1 and $p-1$, which are all relatively prime to p, hence the prime factor p in the numerator cannot be killed by the denominator, proving the claim.

(b) By the binomial theorem, $(x+1)^p = \sum_{k=0}^{p} \binom{p}{k} x^k$, hence part (a) gives that $(x+1)^p \equiv x^p + 1 \pmod{p}$, which means that $x^p + 1$ is a product of p terms.

(c) After reducing $\mathbb{Z}_3[x]$ polynomials (mod $x^3 + 1$), the set of remainders is $\{c_2 x^2 + c_1 x + c_0 : c_0, c_1, c_2 \in \mathbb{Z}_3\}$, hence we have 3^3 polynomials. This set is not a field, because $x + 1$ does not have a multiplicative inverse: part (b) shows that $x+1$ is a factor of x^3+1, hence if we multiply $x+1$ by anything and then reduce (mod x^3+1), the result will be divisible by $x+1$, i.e., cannot be 1.

(d) There is one way to decompose 2 as a sum of strictly smaller positive numbers: $2 = 1+1$. Hence, if x^2+1 is reducible, then it must be the product of two linear polynomials.
The irreducible linear polynomials in $\mathbb{Z}_3[x]$ are $ax+b$, $a = 1, 2$, $b = 0, 1, 2$. Since $\mathbb{Z}_3$ is a field, every nonzero element has a multiplicative inverse, hence, if $f(x) \mid g(x)$, then also $c\, f(x) \mid g(x)$ for any $c \in \mathbb{Z}_3$. This means that it is enough to check $x, x+1, 2x+1, x+2$, and can skip $2x$ and $2x+2$. Checking can be done with long division.
Alternatively, if x^2+1 had a linear factor, then it would also have a root. But $0^2+1 \equiv 1$, $1^2+1 \equiv 2$, $2^2+1 \equiv 2 \pmod 3$, so there is no root.

(e) Say, my number is 994509281. The polynomial in $\mathbb{Z}_3[x]$ is $x^6+2x^5+2x^2+2x+1$. Reducing (mod x^2+1), we get $x^6+2x^5+2x^2+2x+1 = (x^2+1)(x^4+2x^3+2x^2+x)+(x+1)$, so the polynomial in $\mathbb{Z}_3[x]/(x^2+1)$ is $x+1$.
What is its inverse? Euclidean algorithm: $x^2+1 = (x+1)(x+2)+2$, where we can stop because 2 divides 1 in $\mathbb{Z}_3$. This gives $2 = (x^2+1)-(x+2)(x+1)$, so, dividing by 2, which is the same as multiplying by 2, get $1 = 2(x^2+1)+(x+2)(x+1)$. Therefore, the inverse of $x+1$ is $x+2$.

Problem 6. What is the last digit of 7^{7^7} (i.e., 7 to the power 7^7)? **(3 pts)**

Solution. Finding the last digit means we want to find the value (mod 10). So, we can reduce the exponent by $\phi(10) = 4$. So, what is $7^7 \pmod 4$? Here can reduce the base mod 4 and the exponent by $\phi(4) = 2$. So $7^7 \equiv 3^1 \equiv 3 \pmod 4$. So, $7^{7^7} \equiv 7^3 \equiv 3 \pmod{10}$.

A route without Euler's theorem is to notice that $(10a+b)(10a'+b') \equiv bb' \pmod{10}$, hence multiplying the last digits gives the last digit. Therefore, looking at the last digits in 7^k, $k = 1, 2, 3, \ldots$, we see $7, 9, 3, 1, 7$, oops, repetition, so the sequence repeats from here on, it has period 4, so we have to find $7^7 \pmod 4$. By successive squaring, $7 \equiv 3$, $7^2 \equiv 1$, $7^4 \equiv 1 \pmod 4$, so $7^7 \equiv 1 \cdot 1 \cdot 3 \equiv 3 \pmod 4$. So, we have to take the third element of the periodic sequence, which is 3.

Problem 7. Consider $n = 17 \cdot 31 = 527$. How many square roots can a given number have (mod n)? Find an example for each possibility. **(4 pts)**

Solution. By the Chinese Remainder Theorem, $x^2 \equiv y \pmod{527}$ if and only if $x^2 \equiv y$ both (mod 17) and (mod 31). The number of solutions modulo a prime can be 0 (y is not a quadratic residue), 1 ($y \equiv 0$), or 2 (y is a nonzero quadratic residue). So, when combining these using the CRT, we can get $0 \cdot \{0, 1, 2\} = \mathbf{0}$, or $1 \cdot 1 = \mathbf{1}$, or $1 \cdot 2 = \mathbf{2}$, or $2 \cdot 2 = \mathbf{4}$ solutions.

To give examples, one has to find a quadratic non-residue and a non-zero residue mod each prime. Since $x^2 \equiv (-x)^2$, one can list the 8 nonzero residues (mod 17) and the 15 nonzero residues (mod 31) quite quickly:

$$1\ 4\ 9\ 16\ 8\ 2\ 15\ 13 \pmod{17} \quad \text{and} \quad 1\ 4\ 9\ 16\ 25\ 5\ 18\ 2\ 19\ 7\ 28\ 20\ 14\ 10\ 8 \pmod{31}.$$

Thus, for instance, 3 is not a residue modulo either, and 1 and 2 are residues modulo both. Combining them with the CRT, here are four examples:

0: $x^2 \equiv 0 \pmod{17}$ and $x^2 \equiv 3 \pmod{31}$ $\Leftrightarrow$ $x^2 \equiv 34 \pmod{527}$

1: $x^2 \equiv 0 \pmod{17}$ and $x^2 \equiv 0 \pmod{31}$ $\Leftrightarrow$ $x^2 \equiv 0 \pmod{527}$

2: $x^2 \equiv 2 \pmod{17}$ and $x^2 \equiv 0 \pmod{31}$ $\Leftrightarrow$ $x^2 \equiv 155 \pmod{527}$

4: $x^2 \equiv 1 \pmod{17}$ and $x^2 \equiv 1 \pmod{31}$ $\Leftrightarrow$ $x^2 \equiv 1 \pmod{527}$.

(Max possible score: **24 pts**)