Cryptography and Security — Midterm Exam

Serge Vaudenay

1.11.2023

- duration: 1h45
- no documents allowed, except one 2-sided sheet of handwritten notes
- a pocket calculator is allowed
- communication devices are not allowed
- the exam invigilators will **not** answer any technical question during the exam
- readability and style of writing will be part of the grade
- answers should not be written with a pencil

1 Perfect Secrecy with Enigma

This exercise is about Enigma, and the possibility to obtain perfect secrecy when we limit the length of the plaintext. We assume that the Enigma key is uniformly distributed among the Enigma keyspace. We recall that a key is defined by four alphabet permutations $\alpha_0, \beta_0, \gamma_0, \sigma$ and an offset a. The permutations $\alpha_0, \beta_0, \gamma_0$ are elements of a choice of 5 permutations and must be different (they are the rotors). The permutation σ is an involution with 14 fixed points (this is the plugboard). The offset a is an integer such that $0 \leq a < 26^3$. To encrypt a bitstring $x = x_1 \cdots x_m$, we obtain $y = y_1 \cdots y_m$ with

$$y_i = \sigma^{-1} \circ \alpha_{i_1}^{-1} \circ \beta_{i_2}^{-1} \circ \gamma_{i_3}^{-1} \circ \pi \circ \gamma_{i_3} \circ \beta_{i_2} \circ \alpha_{i_1} \circ \sigma(x_i)$$

where π is a fixed involution with no fixed point (this is the reflector), $i+a = 26^2 i_3 + 26 i_2 + i_1$, with $0 \leq i_j < 26$, and $\alpha_i = \rho^i \circ \alpha_0 \circ \rho^{-i}$ (the first rotor in position i), the same for β_i and γ_i, and ρ is the circular rotation of the alphabet (of 26 letters).

Q.1 Recall what is the necessary constraint regarding the message space and the key space to obtain perfect secrecy.

Q.2 What is the minimal message length beyond which this constraint is not respected?
HINT: we have seen in class that the Enigma keyspace has cardinality 2^k with $k \approx 57$.

Q.3 Prove that when we press a key c, the lamp which shows on is never equal to c.

Q.4 Prove that Enigma offers no perfect secrecy even when we limit the message space to one character and the plaintext can be any element of that space with a nonzero probability.

2 Diffie-Hellman Forever

We consider a group G (with multiplicative notations) generated by some g. The textbook Diffie-Hellman protocol is as follows:

- Alice picks a random x, computes $X = g^x$, and sends X to Bob.
- Bob picks a random y, computes $Y = g^y$, and sends Y to Alice.
- Alice computes $K = Y^x$, Bob computes $K = X^y$, and they both use it as a secret key.

Q.1 There are 4 important details missing in the textbook Diffie-Hellman protocol (4 for Alice and 4 for Bob). Spot at least 3.

Q.2 Let p and q be large prime numbers and g be an element of $\mathbf{Z}_p^*$ of order q. Hence, we have $G = \langle g \rangle$. Given a random element $z \in \mathbf{Z}_p^*$, how do we check efficiently if $z \in G$? Analyze the time complexity.

Q.3 Let p, q, r be pairwise different large prime numbers and g be an element of $\mathbf{Z}_p^*$ of order $n = qr$. Hence, we have $G = \langle g \rangle$. Given a random element $z \in \mathbf{Z}_p^*$, how do we check efficiently if $z \in G$? Analyze the time complexity.
HINT: thanks to Bezout, we can write $z = (z^q)^u (z^r)^v$ and use the previous result.

Q.4 Is there any advantage of using a subgroup of $\mathbf{Z}_p^*$ of order qr? Discuss when q and r are known. Discuss when n is known but neither q nor r. Discuss when n is unknown.
[This is an open question. Any answer with a detailed analysis is welcome.]

3 Modulo 99 991

We let $n = 99\,991$ and we want to develop arithmetics modulo n with pen-and-paper.

Q.1 Given a decimal number x of 10 digits, develop an algorithm to reduce it modulo n by using only 2 subtractions and 3 additions of numbers up to 6-digit long. (We take multiplication by 10 as free.)
HINT: $10^5 \bmod 99\,991$.

Q.2 Factor $n - 1$ as a product of prime numbers.

Q.3 How would you verify that n is prime? Estimate the complexity.

Q.4 How to check if x has a square root modulo n, and how to find one when it exists? Estimate the complexity.