Cryptography, homework sheet 4

Due for 2MMC10: 01 October 2020, 10:45 by email to `crypto.course@tue.nl`
and for Mastermath: 29 October 2020, 10:45 by email to `crypto.course@tue.nl`

Team up in groups of two or three to hand in your homework. We do not have capacity to correct all homeworks individually. Do not email Tanja your homework or put homework in mailboxes.
You may use computer algebra systems such as mathematica, gp, or sage or program in C, Java, or Python. Please submit your code (if any) as part of your homework. If you do, make sure that your programs compile and run correctly; the TAs will not debug your programs. The program should also be humanly readable. Code should be sent to `crypto.course@tue.nl`.

1. $3 \in \mathbb{F}_{1013}^*$ generates a group of order 1012, so it generates the whole multiplicative group of the finite field.

   Alice's public key is $h_A = 224$. Use ElGamal encryption to encrypt the message $m = 42$ to her using the "random" nonce $k = 654$.

   You may write the result of exponentiation in one go, without stating intermediate results. No need to show the code.

2. You find two signatures made by Alice. You know that she is using the ElGamal signature scheme over $\mathbb{F}_{2027}$ and that the order of g is $\ell = 1013$, which is prime. The signatures are for $h(m_1) = 345$ and $h(m_2) = 567$ and are given by $(r_1, s_1) = (365, 448)$ and $(r_2, s_2) = (365, 969)$. Compute (a candidate for) Alice's long-term secret a based on these signatures, i.e. break the system.

   Hint: There is a reason that I don't state the generator or Alice's public key; you're not supposed to compute the DL by any other means.

3. $13 \in \mathbb{F}_{1321}^*$ generates a group of order $1320 = 2^3 \cdot 3 \cdot 5 \cdot 11$. Solve the discrete logarithm problem $g = 13, h = 320$ by using the Pohlig-Hellman attack, i.e. find an integer $0 < a < 1320$ such that $h = g^a$ by computing first a modulo $2, 4, 8, 3, 5$, and 11 and then computing a using the Chinese Remainder Theorem.

   Please see the course page for the writeup on how to use Pohling-Hellmann. In particular, you need to use the correct version of the Pohling-Hellman algorithm which updates the target h' in order to reuse the tables with powers of g when you cover higher powers of 2.