

Cryptography, homework sheet 4

Due for 2MMC10: 03 October 2019, 10:45

and for Mastermath: 31 October 2019, 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; my students 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 – but it should be clear what computation you did, e.g., name the algorithm, show the code, give a description.

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 the generator is $n = 1013$. 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.

3. $3 \in \mathbb{F}_{1013}^*$ generates a group of order $1012 = 4 \cdot 11 \cdot 23$. Solve the discrete logarithm problem $g = 3, h = 321$ by using the Pohlig-Hellman attack, i.e. find an integer $0 < a < 1012$ such that $h = g^a$ by computing first a modulo 2, 4, 11, and 23 and then computing a using the Chinese Remainder Theorem.

The *Pohlig-Hellman attack* works in any group and is a way to reduce the hardness of the DLP to the hardness of the DLP in subgroups of prime order. In particular you'll see in the exercise that it works against the DLP in \mathbb{F}_{1013}^* by solving DLPs in groups of size 2, 11, and 23. Here is the general description:

Let G be a cyclic group generated by g and let the challenge be to find $\log_g h = a$. Let the group order n factor as $n = \prod_{i=1}^r p_i^{e_i}$ where $p_i \neq p_j$ for $i \neq j$. Then a can be computed from the information

$$\begin{aligned} a &\equiv a_1 \pmod{p_1^{e_1}} \\ a &\equiv a_2 \pmod{p_2^{e_2}} \\ a &\equiv a_3 \pmod{p_3^{e_3}} \\ &\vdots \\ a &\equiv a_r \pmod{p_r^{e_r}} \end{aligned}$$

by using the Chinese remainder theorem. This is because the $p_i^{e_i}$ are coprime and their product is n . So, if one can find the DL modulo all $p_i^{e_i}$ one can compute the entire DL.

Put $n_i = n/p_i^{e_i}$. Since g has order n the element $g_i = g^{n_i}$ has order $p_i^{e_i}$. The element $h_i = h^{n_i}$ is in the subgroup generated by g_i and it holds that $h_i = g_i^{a_i}$, where $a_i \equiv a \pmod{p_i^{e_i}}$.

E.g. $\mathbb{F}_{16}^* = \langle g \rangle$ has 15 elements, so one can first solve the DLP $h = g^a$ modulo 3 and then modulo 5. For such small numbers one can simply compute h^5 and compare it to $1, g^5$, and g^{10} to find whether a is equivalent to 0, 1, or 2 modulo 3. Then one compares h^3 to $1, g^3, g^6, g^9$, and g^{12} to see whether a is congruent to 0, 1, 2, 3, or 4 modulo 5.

The same approach works also for \mathbb{F}_{17}^* which has $16 = 2^4$ elements – but here one can do much better! Write $a = a_0 + a_1 2 + a_2 2^2 + a_3 2^3$. Then h^8 is either equal to 1 or to $-1 = g^8$ depending on whether a_0 is 0 or 1. Once that result is known we can compare $(h/g^{a_0})^4$ with 1 and -1 to find a_1 etc. So we can solve a much smaller DLP. Instead of going for a modulo $p_i^{e_i}$ at once we can first obtain a modulo p_i , then modulo p_i^2 , then modulo p_i^3 , etc. till $p_i^{e_i}$ by each time solving a DLP in a group of size p_i .

In general, for each p_i in the factorization of n one does the following:

1. Put $h' = h$, $a_{i,-1} = 0$
2. for $j = 0$ to $e_i - 1$
 - (a) put $h' = h' / (g^{a_{i,j-1} p_i^{j-1}})$ //using precomputed g^{-1}
 - (b) solve the DLP of order p_i for $a_{i,j} = \log_{g^{n/p_i}}(h')^{n/p_i^{j+1}}$.

and then combine the $a_{i,j}$ to $a_i = \sum_{j=0}^{e_i-1} a_{i,j} p_i^j$ and then those $a_i \pmod{p_i^{e_i}}$ (using CRT) to $a \pmod{n}$.

Important: the Pohlig-Hellman attack handles one prime at a time, not a prime power. That means that your DL table has only p_i elements and that you solve e_i DLs in subgroups of order p_i . You can see the difference in the example with \mathbb{F}_{17}^* below.

Numerical examples:

$\mathbb{F}_{11}^* = \langle 2 \rangle$, find a so that $3 = 2^a$. So $g = 2$ and $h = 3$. Compute $n_1 = 10/2 = 5$, $g^{n_1} = 2^5 = -1$, and $h^{n_1} = 3^5 = 1$ to see that $a \equiv 0 \pmod{2}$. Then compute $n_2 = 10/5 = 2$, $g^{n_2} = 2^2 = 4$, $g^{2n_2} = 2^4 = 5$, $g^{3n_2} = 2^6 = 9$, and $g^{4n_2} = 2^8 = 3$ and compare that to

$h^{n_2} = 3^2 = 9$ to see that $a \equiv 3 \pmod{5}$. These two congruences imply that $k = a$ and indeed $g^8 = h$.

$\mathbb{F}_{17}^* = \langle 3 \rangle$, find a so that $7 = 3^a$. So $g = 3$ and $h = 7$. In this example we will obtain a one bit at a time. First compare $h^8 = 7^8 = -1$ to 1 and -1 to see that $a \equiv 1 \pmod{2}$. Then compute $h/g = 8$ and then $(h/g)^4 = -1$, so also the next bit is 1 and we see $a \equiv 3 \pmod{4}$. Then compute $h/g^3 = 16$ and then $(h/g^3)^2 = 1$ to see that the next bit is 0, so $a \equiv 3 \pmod{8}$. Finally, since $h/g^3 = 16 = -1$ we see that the highest bit is 1, so $a \equiv 11 \pmod{16}$ and indeed $3^{11} = 7$. This solved the DLP in \mathbb{F}_{17}^* with just 4 very easy computations and comparisons. So computing DLs in fields \mathbb{F}_p with $p = 2^r + 1$ is easy.