Cryptography, homework sheet 5

Due for 2MMC10: 13 October 2016, 10:45

and for Mastermath: 17 November 2016, 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.

1. Majordomo is a program that manages Internet mailing lists. If you send a message to `majordomo@foodplus.com` saying `subscribe recipes`, Majordomo will add you to the `recipes` mailing list, and you will receive several interesting recipes by e-mail every day.

   It is easy to forge mail. You can subscribe a victim, let's say `God@heaven.af.mil`, to the `recipes` mailing list, and thousands more mailing lists, by sending fake subscription requests to Majordomo. `God@heaven.af.mil` will then be flooded with mail.

   Majordomo 1.94, released in October 1996, attempts to protect subscribers as follows. After it receives your subscription request, it sends you a confirmation number. To complete your subscription, you must send a second request containing the confirmation number.

   Majordomo 1.94 generates confirmation numbers as follows. There is a function h that changes strings to numbers. The `recipes` mailing list has a secret string k. The confirmation number for an address a is $h(ka)$. For example, if the secret string is `ossifrage`, and the address is `God@heaven.af.mil`, the confirmation number is h(`ossifrageGod@heaven.af.mil`).

   The function h produces a 32-bit result, computed as follows. Start with 0. Add the first byte of the string. Rotate left 4 bits. Add the next byte of the string. Rotate left 4 bits. Continue adding and rotating until the end of the string.

   Explain how to subscribe `God@heaven.af.mil` to the `recipes` mailing list despite this protection, and explain what Majordomo 1.94 should have done.

2. (No points for this exercise, but do it anyways)
   Last month a bug was found in Signal for Android which meant that in some cases the MAC was over a shorter part of the message, allowing an attacker to append data to a message. More specifically, this bug applied to attachments and came from an error in the code taking a 64-bit value for a 32-bit one. The part that makes this relevant for 2MMC10 is that the implementation used AES in CBC mode. Please read `https://pwnaccelerator.github.io/2016/signal-part2.html`.

3. Here is a toy version of a Wegman-Carter message authentication with which A and B can authenticate t messages: Fix a prime p, e.g. $p = 1000003$. Randomly generate integers $r, s_1, s_2, \ldots, s_t \in \{0, 1, 2, \ldots 1000002\}$. These values are the shared secrets; r is the overall secret and the s_i are per message secrets.

   To authenticate the i-th message m_i the sender expresses m_i in base p as $m_i = m_{i,0} + m_{i,1}p + m_{i,2}p^2 + ... + m_{i,n}p^n$ and computes the authenticator as

   $$a = m_{i,0}r + m_{i,1}r^2 + m_{i,2}r^3 + ... + m_{i,n}r^{n+1} + s_i \bmod p.$$

For simplicity we will do $i = 1$ and omit the extra indices. Compute the authenticator for $m = 45435654243597928347592843 7, r = 483754, s = 342534$.

4. Use the schoolbook version of Pollard's rho method to attack the discrete logarithm problem given by $g = 3, h = 245$ in $\mathbb{F}_{1013}^*$, i.e. find an integer $0 < a < 1012$ such that $h = g^a$, using the t_i and r_i (the twice as fast walk) as defined in class (aand repeated here). Let $t_0 = g, a_0 = 1$, and $b_0 = 0$ and define

$$t_{i+1} = \begin{cases} t_i \cdot g \\ t_i \cdot h \\ t_i^2 \end{cases}, a_{i+1} = \begin{cases} a_i + 1 \\ a_i \\ 2a_i \end{cases}, b_{i+1} = \begin{cases} b_i \\ b_i + 1 \\ 2b_i \end{cases} \text{ for } t_i \equiv \begin{cases} 0 \bmod 3 \\ 1 \bmod 3 \\ 2 \bmod 3 \end{cases},$$

where one takes t_i as an integer. The twice as fast walk has $r_i = t_{2i}$.

The walk could start at any $t_0 = g^{a_0} h^{b_0}$ for known a_0 and b_0 – but then the homework would be harder to correct.

5. Use factor base $\mathcal{F} = \{2, 3, 5, 7, 11, 13\}$ to solve the DLP $h = 281$, $g = 2$, in $\mathbb{F}_{1019}^*$. I.e. pick random powers of $g = 2$, check whether they factor into products of powers of 2,3,5,7,11, and 13; if so, add a relation to a matrix. The columns of the matrix correspond to the discrete logs of 2,3, 5,7,11, and 13. Once you have 6 rows try to solve the matrix; note that these computations take place modulo the group order 1018. It might be that some of the rows are linearly dependent, in that case you need to generate another relation. Once you have all discrete logs of the primes in the factor base, check whether h is smooth and if not find a h/g^i (for some i) which is smooth.

E.g. $2^{291} \equiv 52 \bmod 1019$; over the integers $52 = 2^2 \cdot 13$, so we incluclude the relation $291 \equiv 2a_2 + a_{13} \bmod 1018$. Note that you can run into difficulties inverting modulo 1018 since it is not prime. E.g. $2^{658} \equiv 729 \bmod 1019$; over the integers $729 = 3^6$, so we incluclude the relation $658 \equiv 6a_3 \bmod 1018$ but 6 is not invertible modulo 1018 and we can only determine $a_3 \equiv 449 \bmod 509$ and need to test whether $a_3 = 449$ or $a_3 = 449 + 509$. Here $2^{449} \equiv 1016 \bmod 1019$ and $2^{449+509} \equiv 3 \bmod 1019$, thus $a_3 = 958$.

Hint: if you're using Pari-GP you'll find

```
factor(lift(Mod(2^i,p)))
```

a usefull command.