**Homework sheet 2, due 23 December 2014 at 12:00**

For this exercise sheet you should not use your computer for more functions than a pocket calculator offers you; in particular make sure to give full details when computing inverses and exponentitations.
Submit your homework by encrypted and signed email. This time a different member of your group should handle the submission; ideally you should cc your group members. Do not forget to attach your public key.
Note: the text in 1b was modified to clarify the exercise and make clear that this approach works for a specific value of the last but – which is not necessarily the value of the last block of $M_\ell$. Also, the length of the original message is not explicitly given as is the IV.

1. In SSLv3 one of the two options for symmetric encryption is DES in CBC mode. To protect against message forgery a message authentication code MAC is used. SSLv3 uses the MAC-then-encrypt approach, thus a message $m$ first gets encoded as $M = m||\text{MAC}(m)||\text{pad} = M_1 \dots M_{\ell-1} M_\ell$ and then encrypted using DES with CBC. The padding pad is chosen so that the total length of $M$ is a multiple of 64 (to match the block size of DES) and that the last byte states the length of the padding (including this byte) in bits. Note, the latter means that there always has to be a padding, even if $m||\text{MAC}(m)$ has length a multiple of 64. There are no further requirements on how the padding is chosen. Upon receiving a ciphertext $C$, a computer will decrypt the message $M$, read the last byte to learn the length of the padding to identify $m$ and $\text{MAC}(m)$, and finally verify the MAC. If this verification fails the computer will close the connection.

   (a) Just as a reminder of how CBC works, write how you decrypt the last block of the ciphertext.

   (b) Assume that $C = C_0 C_1 \dots C_{\ell-1} C_\ell$ is a ciphertext so that the $C_\ell$ block comes entirely from the encryption of pad. The first block $C_0$ contains the IV. What is the value of the last byte in $M_\ell$? Show how this gives you a method that for each $0 < i < \ell$ you can test whether the last byte of $M_i$ matches a publicly available value (computed from the $C_i$).

   To give a concrete example let $C_0$ =`01 23 45 67 89 AB CD EF`, $C_{\ell-1}$ =`12 34 56 78 9A BC DE F0` (in hex) and (like above) let $C_\ell$ come entirely from padding. What value of the last byte of $M_1$ can you test for?

2. Users $A, B, C, D$, and $E$ are friends of $S$. They have public keys $(e_A, n_A) = (5, 62857), (e_B, n_B) = (5, 64541), (e_C, n_C) = (5, 69799), (e_D, n_D) = (5, 89179)$, and $(e_E, n_E) = (5, 82583)$. You know that $S$ sends the same message to all of them and you observe the ciphertexts $c_A = 11529, c_B = 60248, c_C = 27504, c_D = 43997$, and $c_E = 44926$. Compute the message.

3. Alice has public key $(e, n) = (3, 262063)$. You capture two messages $c_1 = 156417$ and $c_2 = 6125$ to her and know that the corresponding plaintexts are related as $m_2 = 7m_1 + 19$. Compute the messages $m_1$ and $m_2$.