Modern Cryptology (236506) – Exercise no. 5

Submission in singles until 8/01/2013.

5 bonus points will be given to two sided, printed submissions.

1. (a) Find the modular square roots of 3 modulo 23, using the algorithm we saw in class. Describe your calculations.

   (b) Find the modular square roots of 20 modulo 29, using the algorithm we saw in class. Describe your calculations.

2. Let \( n, e \) be an RSA public key. Suppose that there exist a polynomial time algorithm \( A \) that given \( y = x^e \mod n \) computes the LSB of \( x \). Show how to use \( A \) as a subroutine to construct a polynomial time algorithm \( B \) that solves the RSA problem (i.e., for every \( y = x^e \mod n \) the algorithm \( B \) computes \( x \)). Analyze the running time of \( B \) as a function of the running time of \( A \) and the length of the binary representation of \( n \).

3. In this question we discuss variants of the Rabin method for signing. Let \( p, q \) be large prime numbers, the public key is \( n = pq \). Given a message \( m \in \mathbb{Z}_n \) to sign, we find a square root \( b \) of \( m \), i.e., \( b^2 \equiv m \pmod{n} \). In case such a square root does not exist, we repeatedly concatenate a random string \( r \) to \( m \), and try to find a square root of \( m || r \), until a square root is found.

   Assume that the owner chooses 2 prime numbers \( p \equiv 3 \pmod{8} \) and \( q \equiv 7 \pmod{8} \), calculates \( n = pq \), and publishes \( n \) as his public key for signing. Let \( h = (n + 1)/2 \).

   (a) Show that for all \( m \in \mathbb{Z}_n^* \) exactly one of \( m, -m, hm, -hm \) is a QR modulo \( n \).

      Hint: Show that \(-1\) and \( h \) are QNRs modulo \( p \), that \(-1\) is a QNR modulo \( q \) and that \( h \) is a QR modulo \( q \).

   In Rabin’s method for signing we calculate the square root of the message \( m \). A problem arises when \( m \in \text{QNR}_n \). The following solution is suggested: the signer identifies which of the 4 values \( m, -m, hm, -hm \) is a QR modulo \( n \).

   (b) Explain how.

   Denote this QR by \( \ell \). The signer calculates the square root of \( \ell \) modulo \( n \).

   (c) Explain how.

   And sends one of the roots as the signature.

   (d) How does the signature verification is performed ?

   (e) Explain why such a root is, in fact, a signature on 4 different messages.

      Which messages ?
4. Consider the Zero-Knowledge protocols taught in class.

**Graph Isomorphism**

(a) Explain why the verifier is convinced that the two graphs $G_1$ and $G_2$ are Isomorphic, while the prover never proves it directly.

(b) A simulator for this protocol will gladly supply a transcript ("proving isomorphism") even if $G_1$ and $G_2$ are not isomorphic. Why? and how can it be?

What would happen if it could not generate a transcript in such case?

**Graph Non-Isomorphism**

(c) What will happen if isomorphic $G_1$ and $G_2$ are used in this protocol?

(d) Explain why the verifier is convinced that $G_1$ and $G_2$ are not isomorphic, while the prover is only showing isomorphism of the graph $H$ to $G_1$ or $G_2$.

5.