Institut for Matematik og Datalogi
Syddansk Universitet

November 19, 2019
JFB

Cryptology – 2019 – Assignment 2

Assignment due Thursday, November 21, 23:59

This is assignment must be done individually. Note that this is part of your exam project, and it will count some towards the grade in the course, so you may not work with others. If it is late, it will not be accepted. Turn in the assignment through the SDU Assignment system in Blackboard, and remember to keep your receipt.

You may write your answers in either English or Danish. If you do it by hand, write very neatly and legibly.

1. Explain how the security games for selective forgery for MAC security, SF-PASS and SF-CMA, would differ from that in Figure 11.19 in the textbook for EUF-CMA? (SF stands for Selective Forgery.) Explain.

2. Explain why nonlinearity is important in hash functions.

3. Show all steps in the calculations of the Jacobi symbol $\left(\frac{53}{103}\right)$, using the standard algorithm (using the four properties of the Legendre/Jacobi symbol given in the textbook, but not using any ability to factor other than dividing by 2).

4. Consider the following algorithm for finding square roots modulo a prime $p \equiv 13 \pmod{16}$.

```
procedure SquareRoot($p, a$):
{ Input: A prime $p \equiv 13 \pmod{16}$; a quadratic residue $x \pmod{p}$ }
{ Output: $y$ such that $x \equiv y^2 \pmod{p}$ }

    $q \leftarrow (p-1)/4$
    if ($x^q \equiv 1 \pmod{p}$) then return $x^{(q+1)/2} \pmod{p}$
    else return $x^{(q+1)/2} \cdot 2^q \pmod{p}$
```

(a) Try this algorithm on $p = 61$ and $x = 41$. You do not need to show how you actually do the exponentiation, but show what results you get for other intermediate computations (the larger steps, such as $2^q \pmod{p}$).

(b) Show why the algorithm behaves correctly when it executes the **then** part of the **if** statement.

(c) Consider the result computed in the **else** part of the **if** statement. Show that the result returned there is also correct (for any quadratic residue x modulo $p \equiv 13 \pmod{16}$).

5. Find all square roots of 1 modulo 15. (It is OK to use brute force, but explain what you did do.) Now, use these square roots to factor 15. Do not just use brute force, and explain what you do.

6. Consider running the Miller-Rabin algorithm (Algorithm 2.2 in the textbook) and the Solovay-Strassen algorithm (on lecture note 10) to check the primality/compositeness of 561. A witness to the compositeness of 561 is a value (called a in both algorithms) which makes the algorithm answer that 561 is a composite (and it is the value returned then). Show your work on all parts of this question.

   (a) Is 2 a witness to the compositeness of 561 for the Miller-Rabin algorithm? For the Solovay-Strassen Algorithm?

   (b) Is 3 a witness to the compositeness of 561 for the Miller-Rabin algorithm? For the Solovay-Strassen Algorithm?

   (c) Is 13 a witness to the compositeness of 561 for the Miller-Rabin algorithm? For the Solovay-Strassen Algorithm?

   (d) Suppose that a is a not a witness to the compositeness of 561 for Miller-Rabin, but $a^{35} \not\equiv 1 \pmod{561}$. Prove that a is a quadratic nonresidue modulo 561. (Let me know if you want a hint.)

   (e) Find a quadratic nonresidue $a \pmod{561}$, which has Jacobi symbol $\left(\frac{a}{561}\right) = 1$. Is it a witness using either Miller-Rabin or Solovay-Strassen?