Home Assignment 3

Advanced Web Security

2017

B-assignments

For grade 3, complete the three B-assignments below and solve them in groups of two students.

B-1 Assume that we have a commitment scheme $x = h(v, k)$, where v is a 1-bit commitment, k is a K-bit random string and h is a hash function with output truncated to X bits. Fix K to be 16 bits. For different appropriate choices of X, simulate

a) The probability of breaking the binding property of the scheme.

b) The probability of breaking the concealing property of the scheme.

What can you say about how the probabilities varies with X? Make sure you clearly present your algorithms for computing the probabilities.

Assessment:

- Summarize your work in a short report, making it clear that the program works as intended.
- Upload your report to Urkund, paul.stankovski.lu@analys.urkund.se. It is enough that one student per group does this.
- Upload your report to Moodle (it will be manually graded). Both students must do this.

B-2 After winning the election, Mrs. Flinton is now in charge of the military weaponry, i.e. the nuclear launch codes. Being that Mrs. Flinton is incapable of handling technology, she accidentally launched nuclear weapons against a foreign country. To stop a plausible World War III, you have to stop the missiles. The deactivation process utilizes a (k,n) threshold scheme, and you need to provide the master secret of this scheme to deactivate the nuclear launch.

Implement a program that takes as input

- parameters k and n with $3 \leq k < n \leq 8$,
- your private polynomial,
- polynomial shares from collaborating participants.

The program output should be the deactivation code (an integer).

Example:
You are participant 1 out of 8 in a (5,8) threshold scheme. All participants have each chosen a private polynomial of degree 4. The secret master polynomial is simply the sum of all your individual private polynomials, so that

$$f(x) = f_1(x) + f_2(x) + ... + f_n(x),$$

and the master secret is the constant term (an integer) of this polynomial.

Your private polynomial is $f_1(x) = 13 + 8x + 11x^2 + 1x^3 + 5x^4$.
You have generously shared points on your polynomial, one with each other participant;

$$\begin{aligned} f_1(2) &= 161, \\ f_1(3) &= 568, \\ f_1(4) &= 1565, \\ f_1(5) &= 3578, \\ f_1(6) &= 7153, \\ f_1(7) &= 12956, \\ f_1(8) &= 21773. \end{aligned}$$

You have also been given shares from the other participants' polynomials, one from each participant;

$$\begin{aligned} f_2(1) &= 75, \\ f_3(1) &= 75, \\ f_4(1) &= 54, \\ f_5(1) &= 52, \\ f_6(1) &= 77, \\ f_7(1) &= 54, \\ f_8(1) &= 43. \end{aligned}$$

Collaborating with participants 2, 4, 5 and 7, they reveal their points on the master polynomial to you;

$$\begin{aligned} f(2) &= f_1(2) + ... + f_8(2) = 2782, \\ f(4) &= f_1(4) + ... + f_8(4) = 30822, \\ f(5) &= f_1(5) + ... + f_8(5) = 70960, \\ f(7) &= f_1(7) + ... + f_8(7) = 256422. \end{aligned}$$

The deactivation code in this case is 110.

Assessment:

- Upload your code to Urkund, paul.stankovski.lu@analys.urkund.se. One upload per group is sufficient.
- There will be one Moodle question following the problem statement above. **Both students must finish the Moodle quiz**. There will be a test quiz on Moodle, where you can try your implementation as many times as you like. The test quiz will not be graded.

B-3 Several topics and problems encountered in the course so far are related to the notions of *semantic security* and *malleability* of cryptosystems. This is also related to the ciphertext indistinguishability properties of cryptosystems, i.e., *IND-CPA*, *IND-CCA* and *IND-CCA2*. Read about these different notions and understand their properties, definitions and how they apply to RSA and ElGamal encryption. You are free to use any source you wish, but the Wikipedia entries are very good and sufficient for our purpose.

Assessment:

- There will be one Moodle-question with statements regarding the different properties. Four of these statements are correct and your task is to identify these statements. You will receive 0.5p for each correct answer and -0.5p for each wrong answer. You can never get less than 0.0p. It can be a good idea to have the material you have used accessible when you answer the questions, but it is extremely recommended that you read and understand it in advance. Both students must finish the quiz.

C-Assignments

For grade 4, complete the C-assignment below and solve it in groups of two students.

C-1 In an upcoming presidental election, the inhabitants can vote on one of the candidates – Grump and Flinton. The voting system is a new e-voting system deployed by your company. Being the security engineer, it is your job to implement a function that counts the votes. The votes are encrypted using the Paillier cryptosystem[1]. Since the votes are anonymous, you are not allowed to decrypt a single vote. Rather, you have to utilize the homomorphic property of Paillier:

$$E(v_1) \cdot E(v_2) = E(v_1 + v_2)$$

to get the total sum of votes. A vote for Mr. Grump is encoded as a +1, whereas a vote for Mrs. Flinton is encoded as -1. If the sum of all votes are positive, Mr. Grump wins and vice versa. Note that in $\mathbb{Z}_n$, the number "$-x$" is written as "$n - x$".

Implement a program that takes as input

- Two prime numbers p and q.
- An element $g \in \mathbb{Z}^*_{n^2}$.
- A file containing the encrypted votes, one per line.

The program output should be the sum of the votes, e.g. 5, -3.

Example:
We have three voters and all voted for Mrs. Flinton. The primes used are $(p, q) = (5, 7)$, $g = 867$, and the following (integer) ciphertexts:

```
929
296
428
```

The product of ciphertexts is $c = c_1 \cdot c_2 \cdot c_3 = 52 \pmod{n^2}$ which gives the sum of votes, $v_{tot} = 32 = -3 \pmod{n}$

Note: plaintexts are reduced mod n, ciphertexts are reduced mod n^2.

Assessment:

- Upload your code to Urkund, paul.stankovski.lu@analys.urkund.se. One upload per group is sufficient.
- There will be one Moodle question following the problem statement above. **Both students must finish the Moodle quiz**. There will be a test quiz on Moodle, where you can try your implementation as many times as you like. The test quiz will not be graded.

[1] `https://en.wikipedia.org/wiki/Paillier_cryptosystem`