Home Assignment 3

Advanced Web Security

2019

B-assignments

For grade 3, complete the three B-assignments below and solve them in groups of ≤ 2 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 the following.

1) The probability of breaking the binding property of the scheme. Note that we assume a non-idiot attacker. This means that the attacker optimizes her chances of success by making sure to choose a value for k that she knows can break the binding property *before* she commits to v. This attack is closely related to a well-known property for hash functions.
2) The probability of breaking the concealing property of the scheme. Here the attack is carried out by the receiver of the commitment and it is thus not possible to choose optimum values before the commitment is made. This attack is also closely related to a well-known property for hash functions.

For clarity, we define the probablity of breaking the concealing property as the ratio between the number of x values for which the committed bit can be uniquely determined and the total number of possible x values.

Write a short report that summarizes your work and your findings. Make sure that your report includes at least the following aspects.

- Describe in general terms how an attacker would proceed to perform the two attacks.
- Clearly present your algorithms for computing the probabilities.
- What can you say about how the probabilities varies with X? Plot graphs illustrating the probabilities.
- Identify the hash function properties that the two attacks relate to. What is the relation? Treat both cases separately. A hint is that the birthday paradox is closely related to one of the two cases.

To get on the right track, it can be useful to consider two column vectors, representing all possible values for x:

$v = 0$	$v = 1$
$h(0, 0)$	$h(1, 0)$
$h(0, 1)$	$h(1, 1)$
$h(0, 2)$	$h(1, 2)$
$\vdots$	$\vdots$
$h(0, 2^{16} - 1)$	$h(1, 2^{16} - 1)$

Now, what is required for breaking the binding property? What is required for breaking the concealing property?

Assessment:

- Upload your report to Urkund, jonathan.sonnerup.lu@analys.urkund.se. Only one student per group must do this.
- Upload your code to Urkund, jonathan.sonnerup.lu@analys.urkund.se. Only one student per group must do this.
- Upload your report to Moodle (it will be manually graded). Both students must do this.

B-2 A recently elected, semi-competent, leader of a large country accidentally initiated a nuclear missile launch. Your task is to prevent disaster by deactivating the detonation circuit. The deactivation process utilizes a (k,n) threshold scheme, and you need to provide the master secret of the scheme.

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, jonathan.sonnerup.lu@analys.urkund.se. Only one student per group must do this.
- There will be a 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-CCA1* and *IND-CCA2*. Read about these different notions and understand their properties, definitions and how they apply to RSA and, in particular, ElGamal encryption. You are free to use any source you wish, but the Wikipedia entries are very good and sufficient for our purpose. (Note that some texts only use CCA nowadays when they actually mean CCA2. Notations sometimes do change over time when things evolve. The *ciphertext indistinguishability* page on wikipedia is quite clear though.)

Assessment:

An individual oral examination will be used for assessment of this assignment. There will be a link to a Doodle on Moodle, on which you sign up for a time slot. To simplify things, you will even get the exact questions given to you on the examination. They are the following:

1. Argue that El-Gamal is IND-CPA. You do not have to give a formal proof, but you should clearly argue for it (using the mathematical expressions defining El Gamal).
2. Show that El-Gamal is malleable.
3. Show that El-Gamal is not IND-CCA2.

The above obviously assumes that you know the definitions of the three properties (in case you are asked complementary questions about them during the examination) and understands the answers. You will have exactly 5 minutes to provide the answers, using a whiteboard. Do not bring any notes to copy the answers from. We already know that you can read. One last thing: Be on time. Examination schedule will be tight.

C-Assignments

For grade 4, complete the C-assignment below and solve it in groups of ≤ 2 students.

C-1 In a presidential election, the people can vote on one of two candidates. Let us call them 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 is 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 we have the following (integer) ciphertexts:

```
929
296
428
```

The product of the 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, jonathan.sonnerup.lu@analys.urkund.se. Only one student per group must do this.
- There will be a 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