KTH Computer Science and Communication

Homework II, Foundations of Cryptography 2015

Before you start:

1. The deadlines in this course are strict. This homework set is due as specified at `https://www.kth.se/social/course/DD2448/subgroup/vt-2015-60028/page/deadlines-9`.
2. Read the detailed homework rules at `https://www.kth.se/social/files/54b92449f276547e23765898/solution_rules.pdf`.
3. Read about I and T-points, and how these translate into grades, in the course description at `https://www.kth.se/social/files/54baa2cbf276547f11c5e721/course_description.pdf`.
4. You may only submit solutions for a nominal value of 50 points in total (summing I and T points).

The problems are given in no particular order. If something seems wrong, then visit `https://www.kth.se/social/course/DD2448/subgroup/vt-2015-60028/page/handouts-8` to see if any errata was posted. If this does not help, then email `dog@csc.kth.se`. Don't forget to prefix your email subject with `Krypto15`.

We may publish hints on the homepage as well if a problem appears to be harder than expected.

Preliminaries

Definition 1 *The **Discrete Logarithm (DL)** assumption in a group G of prime order q states that if g and y are randomly chosen in G, then for every polynomial time algorithm A, $\Pr\left[A(g,y) = \log_g y\right]$ is negligible.*

Definition 2 *The **Diffie-Hellman (DH)** assumption in a group G of prime order q with generator g states that if $a, b \in \mathbb{Z}_q$ are randomly chosen, then for every polynomial time algorithm A, $\Pr\left[A(g^a, g^b) = g^{ab}\right]$ is negligible.*

Definition 3 *The **Decision Diffie-Hellman (DDH)** assumption in a group G of prime order q with generator g states that if $a, b, c \in \mathbb{Z}_q$ are randomly chosen, then for every polynomial time algorithm A,*
$\left|\Pr\left[A(g^a, g^b, g^{ab}) = 1\right] - \Pr\left[A(g^a, g^b, g^c) = 1\right]\right|$ *is negligible.*

Recall the notational conventions we discussed in class. The group G is for example formally a family $G = \{G_{q_n}\}$ of groups of prime orders q_n with $\lfloor \log_2 q_n \rfloor = n$, $g = \{g_n\}$ where g_n is a generator of G_{q_n}, and $q = \{q_n\}$. Thus, above we abuse notation and remove the index n from our notation, i.e., it is understood that G, q and g really means G_n, q_n, and g_n for increasing n.

Problems

1 (3I) Implement modular exponentiation from modular multiplication. A detailed description is found on Kattis. `https://kth.kattis.com/problems/kth:krypto:modexp`. Make sure that your code is commented and well structured. Up to 3I points may be subtracted if this is not the case. Keep in mind that you must be able to explain your solution during the oral exam.

2 (3I) Implement Chinese remaindering. A detailed description is found on Kattis. `https://kth.kattis.com/problems/kth:krypto:crt`. Make sure that your code is commented and well structured. Up to 3I points may be subtracted if this is not the case. Keep in mind that you must be able to explain your solution during the oral exam.

3 (5I) Compute the factorization of an RSA modulus from its encryption and decryption exponents. A detailed description is found on Kattis. `https://kth.kattis.com/problems/kth:krypto:rsafact`. Make sure that your code is commented and well structured. Up to 5I points may be subtracted if this is not the case. Keep in mind that you must be able to explain your solution during the oral exam.

4 (2I) Determine basic properties of elliptic curves. A detailed description is found on Kattis. `https://kth.kattis.com/problems/kth:krypto:ellipticcurvepoints`. Make sure that your code is commented and well structured. Up to 2I points may be subtracted if this is not the case. Keep in mind that you must be able to explain your solution during the oral exam.

5 (8T) Do a literature study of the (in)security of the SSL/TLS protocol (different versions through the years). Summarize on roughly a page[1] your findings in your own words. Summarizing SSL/TLS and issues with it in one page is challenging. Both your choice of content (5T) and the presentation (3T) is judged. Thus, choose carefully what you present and make sure you have time to proof read and revise your text before submitting.

6 The goal of this problem is to *prove* the following implications covered in class. In other words, the difficulty in solving this problem is not understanding that the implications hold, but to write down a *rigorous* proof. Thus, in this particular problem, any handwaving give zero points. A proof consists of a description of an efficient reduction and a mathematical analysis thereof.

6a (2T) Prove that the DH assumption implies the DL assumption.

6b (2T) Prove that the DDH assumption implies the DH assumption.

7 Suppose you need to generate an n-bit prime p_0 of the form $p_0 = 2p_1 + 1$, where $p_1 = 2p_2 + 1$ and p_2 are primes.

7a (6T) Describe an algorithm and perform a heuristic analysis of its expected running time. Implement your algorithm and compare your practical results with your theoretical analysis.

7b (2T) What happens if we relax the problem and replace 2 in each equality by an integer $1 < k < 2\sqrt{n}$ of your own choice?

For each case you must report your results in the form of a plot for increasing values of n.

[1] Use your own judgement to figure out what a page is.

8 (5T) Let $p = kq + 1$ and q be primes such that $\log q = n$, $\log k = n$ and such that the bit size of every prime factor of k is bounded by $\log n$. Let g be a generator of the unique subgroup of $\mathbb{Z}_p^*$ of order q.

I pick $x \in \mathbb{Z}_q$ randomly and hand you $y = g^x$. Then you may ask me any number of questions of the form $u \in \mathbb{Z}_p^*$, which I answer by $u^x \bmod p$. Explain how you can compute x efficiently (describe your algorithm and analyze its running time).

9 Let $\mathsf{CS} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a public key cryptosystem. More precisely:

- Gen is a probabilistic key generation algorithm that on input 1^n (security parameter n in unary representation) outputs a key pair. We denote this by $(pk, sk) = \mathsf{Gen}(1^n)$.
- Enc is an encryption algorithm that takes a public key pk, a message $m \in \{0,1\}^n$, and randomness $r \in \{0,1\}^n$ as input and produces a ciphertext. We denote this by $\mathsf{Enc}_{pk}(m, r)$.
- Dec is a decryption algorithm that takes a secret key sk and a ciphertext c as input and outputs the plaintext. We denote this by $m = \mathsf{Dec}_{sk}(c)$.

Denote by $\mathsf{CS}^k = (\mathsf{Gen}^k, \mathsf{Enc}^k, \mathsf{Dec}^k)$ the cryptosystem defined as follows:

- Gen^k is identical to Gen
- Enc^k takes a public key pk, a message $m \in \{0,1\}^{n \times k}$, and randomness $r \in \{0,1\}^{n \times k}$ as input and outputs $(\mathsf{Enc}_{pk}(m_1, r_1), \ldots, \mathsf{Enc}_{pk}(m_k, r_k))$.
- Dec^k takes a secret key sk and a ciphertext $c = (c_1, \ldots, c_k)$ as input and outputs a plaintext $(\mathsf{Dec}_{sk}(c_1), \ldots, \mathsf{Dec}_{sk}(c_k))$.

9a (7T) Prove that if CS is semantically secure, then CS^2 is semantically secure.

9b (3T) Prove that for every polynomial $k(n)$, if CS is semantically secure, then $\mathsf{CS}^{k(n)}$ is semantically secure.

The solutions to these problems were covered on the blackboard in class, so you need to be **more rigorous** than that! Imagine that your life depended on convincing your worst enemy.

10 (2T) We denote the Legendre/Jacobi symbol of a modulo b by $\left(\frac{a}{b}\right)$. For each of the following symbols, (1) state if it is a Legendre or Jacobi symbol, (2) determine if the symbol is defined on the given inputs, and (3) *compute the symbol by hand* (I want to see your intermediate results) if possible: $\left(\frac{26}{97}\right)$, $\left(\frac{17}{995}\right)$, $\left(\frac{39}{14}\right)$, $\left(\frac{562045798654}{754835}\right)$, and $\left(\frac{75456}{574565923786}\right)$.