KTH Computer Science and Communication

Homework II, Foundations of Cryptography 2016

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-2016-krypto16/page/deadlines-16`.
2. Read the detailed homework rules at `https://www.kth.se/social/files/5686fcd8f276542387729c18/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/5692df7bf2765405aca1825f/course_description.pdf`.
4. You may only submit solutions for a nominal value of 50 points in total (summing I and T points). The total number of points below may be larger and this should be interpreted as giving you a way to choose problems you like.

The problems are given in no particular order. If something seems wrong, then visit `https://www.kth.se/social/course/DD2448/subgroup/vt-2016-krypto16/page/handouts-10` to see if any errata was posted. If this does not help, then email `dog@kth.se`. Don't forget to prefix your email subject with `Krypto16`.

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

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 (2T) In some applications side channel attacks are a concern. Describe what a side channel attack is. Find as many side channels as possible that as been exploited in the research literature. Cite each relevant paper properly in your answer with a brief description in a few sentences.

6 The goal of this problem is to study the OpenSSL source code to get feeling for what real world code for cryptography can look like.

6a (1T) Identify and report the path to the file containing the optimized implementation of P-256.

6b (1T) Determine if any bugs have been fixed in this code since it was committed to the code base.

6c (3T) Describe the technique used in the implementation of P-256 to counter side channel attacks.

7 The generic elliptic curves covered in class uses separate code for doubling, adding, and treatment of the point at infinity. For some curves this is not necessary, i.e., there is no need for special code.

7a (3T) Dan Bernstein has published several papers about this. Read enough to be able to explain the key ideas.

7b (2T) What is the advantages of such curves in practice?

8 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.

8a (5T) 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.

8b (2T) What happens if we relax the problem and replace 2 in each equality by integers $1 < k_1, k_2 < 2^{\sqrt{n}}$ of your own choice (i.e., you must find any p_0, p_1, p_2 and k_1, k_2)?

9 (5T) Read the paper by Lenstra et al. `http://eprint.iacr.org/2012/064.pdf`. Summarize on roughly a page the findings in this paper in your own words. Make sure that you do not submit a solution unless you actually read the paper.

10 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 $u^x \bmod p$.

10a (2T) Explain how you can compute x efficiently (describe your algorithm and analyze its running time).

10b (1T) What is the important lesson to learn from this example? (This was mentioned in class, but you will not find it on any slides.)

10c (1T) How would you address this problem in an implementation of the protocol?

11 The goal of this problem is that you write out the details of a proof in cryptography on your own. We have already covered this result in class. 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 $(\mathsf{Dec}_{sk}(c_1), \ldots, \mathsf{Dec}_{sk}(c_k))$.

11a (7T) Prove that if CS is secure, then CS^2 is secure.

11b (3T) Prove that for every polynomial $k(n)$, if CS is secure, then $\mathsf{CS}^{k(n)}$ is secure.

You need to be **more rigorous** than what we did in class that! Imagine that your life depended on convincing your worst enemy.

12 (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: $\left(\frac{15}{28}\right)$, $\left(\frac{19}{357}\right)$, $\left(\frac{39}{33}\right)$, $\left(\frac{598120457575754}{75456831}\right)$, and $\left(\frac{57384}{53475546935698}\right)$.