KTH Computer Science and Communication

Homework III, 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

Problems

1 (12I) Implement the arithmetic of an elliptic curve. A detailed description is found on Kattis. `https://kth.kattis.com/problems/kth:krypto:ellipticcurvearithm`. Make sure that your code is commented and well structured. Up to 12I 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 (10I) Implement the SHA-256 hash function. A detailed description is found on Kattis. `https://kth.kattis.com/problems/kth:krypto:sha256`. Feel free to read from different sources on how to make an efficient implementation, but any borrowed ideas should be explained briefly in the solutions submitted on paper. You must also be prepared to explain in detail what you did and why at the oral exam. Make sure that your code is commented and well structured. Up to 10I points may be subtracted if this is not the case.

3 In this problem we investigate how to parallelize computations of hash functions. Denote by $H : \{0,1\}^{2n} \to \{0,1\}^n$ a collision resistant hash function (think of it as chosen randomly from a family). In class we covered the Merkle-Damgård construction, but this is inherently sequential in nature. We also discussed some proposals from students that should be useful in this problem.

Your job is to construct a distributed and parallelized collision resistant hash function that is parameterized by the performance characteristics of a set of heterogeneouos computing devices with various number of cores. The goal is of course that the result is as fast as possible.

3a (8T) Your algorithm must take a list of the form $\big((f_{1,1},\ldots,f_{1,l_1}),\ldots,(f_{1,t},\ldots,f_{t,l_t})\big)$, where $f_{i,j}$ is the frequency of the jth core of the ith computing device, as input and output a distributed program $\big((p_{1,1},\ldots,p_{1,l_1}),\ldots,(p_{1,t},\ldots,p_{t,l_t})\big)$, where $p_{i,j}$ is the program for the jth core of the ith device.

Make reasonable assumptions on the bandwidth of the network, the available memory, and any other parameters you find relevant, depending on the speed of each device and keep in mind that you can not hog all of it, since there are other applications.

Note that you can not assume that the hash function is lightning fast on all devices, e.g., we could be dealing with a sensor network where each device is quite weak.

3b (3T) Estimate the speed of your construction and argue that it is "optimal" as far as reasonably simple schemes go, i.e., we are looking for a back-of-the-envelope calculation and not a rigorous proof here.

3c (5T) Prove rigorously that your construction is collision resistant. (Given what we covered in class and the solution in the first subproblem this should be straight forward.)

Comment: There is plenty of room for interpretation and making different reasonable assumptions in this problem. If you would be willing to defend your assumptions to a future employer, then they are probably good.

4 ECRYPT II Yearly Report on Algorithms and Keysizes (2011-2012) `http://www.ecrypt.eu.org/documents/D.SPA.20.pdf` discusses the security levels of various cryptographic primitives. Keep in mind that you must be prepared to elaborate on your solution during the oral exam.

4a (4T) Give a one-page high level of the approach used to derive the recommended key sizes. Your target audience are fellow students that did not solve this problem.

4b (2T) Argue in the same way as the ECRYPT II project and fill in additional rows in Table 7.2 and Table 7.3. You must motivate your solution.

4c (1T) What do you think about the future of cryptography based on factoring versus cryptography based on the discrete logarithm assumption in elliptic curve groups?

5 You are given an odd integer n-bit composite modulus N and two elements $g, y \in \mathbb{Z}_N^*$, where $y = g^x \bmod N$, and then we execute the following protocol:

1. I choose $r \in [0, 2^{2n} - 1]$ randomly and compute $\alpha = g^r \bmod N$.
2. You choose a challenge $c \in [0, 2^n - 1]$ and hand it to me.
3. I reply by $d = xc + r$ (computed over the integers without any modular reduction).
4. You accept if $y^c \alpha = g^d \bmod N$.

In class we considered a similar protocol executed over a group of prime order and argued that this is a so called proof of knowledge.

5a (1T) Prove that if both me and you follow the protocol, then you accept.

5b (3T) Describe what goes wrong if we use the analysis covered in class for the above protocol?

5c (2T) Can you still say something useful about this protocol? Points for interesting ideas!

Comment: It turns out that no protocol from a large natural class of generalizations of the above can be a proof of knowledge. I assume that this makes you cry, since the protocol is so beautiful, but negative results in cryptography often simply means "we have to do it in another way" or "the security requirements are unnecessarily strict", so you can wipe the tears from your eyes and solve the next problem.

6 This problem investigates Lamport's one-time signatures.

6a (2T) Describe how Lamport's one-time signatures work.

6b (1T) Suppose that Lamport's signature scheme is used more than once for the same public and secret keys. Why is this dangerous?

Comment: One-time signatures may seem useless, but sometimes this is all we need and Lamport's signatures are beautiful in their simplicity.