KTH Computer Science
and Communication

Homework IV, 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.

Problems

1 (10I) Implement the recovery phase of Feldman's verifiable secret sharing scheme. A detailed description is found on Kattis. `https://kth.kattis.scrool.se/problems/feldman`. Make sure that your code is commented and well structured. Up to 10I 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 In this problem we investigate side channel attacks.

2a (2T) Summarize what side channel attacks are and why they are important to consider when implementing cryptographic primitives.

2b (3T) Describe three different side channels and their characteristics.

2c (5T) Investigate the optimized implementations of the standard elliptic curves P-224, P-256, and P-521 in OpenSSL. Give the paths to the relevant files in the repository and describe what is done in this code to avoid side channel attacks.

Definition 1 *The* ***Strong RSA*** *assumption states that if $N = pq$, where p and q are randomly chosen primes with the same number of bits and g is randomly chosen in $\mathbb{Z}_N^*$, then for every polynomial time algorithm A,* $\Pr[A(N, g) = (e, \beta) \wedge \beta^e = g \bmod N \wedge e > 1]$ *is negligible.*

3 Consider the hash function defined as follows. Let $N = pq$ where p and q are randomly chosen safe primes of the same bit-size, and let g be randomly chosen in $\mathbb{Z}_N^*$ with order $(p-1)(q-1)/4$. Then define $h_{N,g}(x) = g^x \bmod N$ for $x \in \mathbb{Z}$.

3a (4T) Prove that a multiple of $(p-1)(q-1)/4$ can be computed from a collision.

3b (2T) Use this fact to prove that the hash function is collision-resistant under the strong RSA assumption.

4 In this problem we investigate secret sharing schemes. Please read about the basics on the Internet.

In class we considered the following secret sharing scheme. To share a secret $s_0 \in \{0, 1\}^n$, the dealer chooses $s_1, s_2 \in \{0, 1\}^n$ randomly under the restriction that $s_1 \oplus s_2 = s_0$. The ith receiver gets s_i and the two receivers can recover s_0 by forming $s_0 = s_1 \oplus s_2$.

4a (1T) Generalize the system to k receivers for any positive k.

4b (3T) Prove that in your scheme, $k - 1$ shares give no information about the secret. Hint: Use ideas from the analysis of the one time pad cryptosystem covered in class.

4c (3T) Let $R = \{P_1, \ldots, P_k\}$ be a set of receivers. Consider a set $A = \{T_1, \ldots, T_s\}$ of sets $T_i \subset R$. A secret sharing scheme for such a set must have the property that: (1) for every set $T \in A$ of receivers their secret shared can be used to recover the secret, and (2) for every set $T \notin A$ of receivers the list of their secret shares give no information about the secret.

Characterize sets of the form A for which there can not be any secret sharing scheme.

You may only, and need only, argue abstractly in terms of the sets themselves, i.e., it does not help to read about concrete secret sharing schemes to solve this problem.

5 Consider mobile networks where the communication to and from each mobile phone is encrypted.

5a (2T) Explain how far up the network the communication remains encrypted, i.e., identify the entity that decrypts data encrypted by the mobile phone. Describe this for two generations of systems for which this aspect differs.

5b (2T) Describe pros and cons with each approach.

6 In this problem we develop a pseudo-random generator which is provably secure under the DDH assumption. Let $p = 2q + 1$ be a safe prime with $\lfloor \log_2 q \rfloor = n$ and let G_q be the subgroup of quadratic residues in $\mathbb{Z}_p^*$.

6a (4T) Suppose that u is a random element in G_q. Prove that $u' = \min\{u, p - u\} \bmod q$ is a randomly distributed element in $\mathbb{Z}_q$, i.e., to generate a random element in $\mathbb{Z}_q$ we can pick a random element u in G_q and then output u or $p - u$ modulo q depending on which is smallest.

6b (2T) We say that $f : G_q \times \mathbb{Z}_2^2 \to G_q^{t(n)}$ is a pseudo-random G_q-generator if $t(n) > 10$ and for every polynomial time algorithm A

$$\left| \Pr_{(g,x,r) \in G_q \times \mathbb{Z}_q^2} \left[A\big(f(g, x, r)\big) = 1 \right] - \Pr_{u \in G_q^{t(n)}} \left[A(u) = 1 \right] \right|$$

Use the first subproblem to construct a pseudo-random G_q-generator f.

Hint: Iterate the map $f_{g,x}(r) = (g^r, g^{xr})$ and in each iteration output some of the output and keep the rest for the next iteration.

6c (5T) Prove that your function f is indeed a pseudo-random G_q-generator.

Hint: Use a hybrid argument similar to what we did in class for the PRG based on a pseudo-random permutation to replace an increasing number of the iterations of your construction by true randomness.

6d (3T) Let $N_1 > N_2$, let u_1 be randomly distributed in $\mathbb{Z}_{N_1}$, and define $u_2 = u_1 \bmod N_2$. Prove that $\sum_{s \in \mathbb{Z}_{N_2}} |\Pr[u_2 = s] - 1/N_2| < O(N_2/N_1)$, i.e., we can convert a random element in $\mathbb{Z}_{N_1}$ to an almost random element in $\mathbb{Z}_{N_2}$ if N_2 is small enough compared to N_1. (Here you should think of both N_1 and N_2 as growing with the security parameter n, but N_1 grows faster than N_2.)

6e (3T) Explain how to combine the above subproblems to construct a pseudo-random generator.