DD2448 Foundations of Cryptography (krypto25)
Homework

Douglas Wikström, dog@kth.se

May 9, 2025

Abstract

Make sure that you read and understand `Files→Homework/solution_rules.pdf` at Canvas before you start. This document details the rules for solving and handing in your solutions.

This homework has 50 T points in total. Problems appear in no particular order.

Please consult the most recent version of this document at Canvas before you contact us regarding something you think is wrong or should be clarified. We post versions with corrections or clarifications if necessary.

Problem 1 (Negligible Functions). Let $l(n)$ be a polynomial and let $\epsilon(n)$ be a negligible function.

Task 1.1 (1T). Prove that $l(n)\epsilon(n)$ is negligible in the parameter n.

Task 1.2 (1T). Consider a sequence of binary random variables $X_{n,1}, \ldots, X_{n,l(n)}$ such that $\Pr[X_{n,i} = 1] \leq \epsilon(n)$. Prove that $\Pr[\sum_{i=1}^{l(n)} X_{n,i} > 0]$ is negligible in the parameter n.

"If each bad event occurs with negligible probability and we have polynomially many events, then the probability that any bad event occurs is also negligible."

Problem 2 (Non-negligible Functions). Let $l(n)$ be a polynomial and let $\Delta(n)$ be a non-negligible function.

Task 2.1 (1T). Prove that $\Delta(n)/l(n)$ is non-negligible in the parameter n.

Task 2.2 (1T). Consider a sequence of binary random variables $X_{n,1}, \ldots, X_{n,l(n)}$ such that $\Pr[\sum_{i=1}^{l(n)} X_{n,i} > 0] \geq \Delta(n)$. Prove that there exists an index $i(n)$ such that $\Pr[X_{n,i(n)} = 1]$ is non-negligible in the parameter n.

"If something bad occurs with non-neglible probability and we have polynomially many events, then there is at least one specific bad event that occurs with non-negligible probability."

Problem 3 (Feistel Ciphers). Let $E_t : \{0,1\}^{tn} \times \{0,1\}^n \to \{0,1\}^n$ be an n-bit block cipher with tn-bit keys, consisting of a t-round Feistel network. Let "$\|$" denote concatenation and let f_i be the ith Feistel function. Then denote the key by $k = k_1\|k_2\|..\|k_t$, the plaintext by $L_0\|R_0 \in \{0,1\}^n$, and the output in round $s \geq 1$ by $L_s\|R_s$, i.e., the output ciphertext is $L_t\|R_t$. Assume that $f_i(k_i, \cdot)$ is pseudo-random function for a random k_i.

We consider the strong definition of pseudo-random permutations, where the adversary has both an encryption and a decryption oracle (as in the lecture slides).

Task 3.1 (1T). Draw the Feistel network for $t = 1, 2, 3$.

Task 3.2 (2T). Show that if $t = 1$, then the Feistel network is not a pseudorandom permutation.

Task 3.3 (4T). Show that if $t = 2$, then the Feistel network is not a pseudorandom permutation.

Task 3.4 (8T). Show that if $t = 3$, then the Feistel network is not a pseudorandom permutation. (Hint: Look at several related inputs and outputs. Evaluate the permutation as well as its inverse on these.)

Problem 4 (CPA Security of El Gamal Under DDH Assumption). Recall the El Gamal cryptosystem. We let G_q be a group of prime order q with generator g, and a keypair (y, x) is generated by choosing a secret key $x \in \mathbb{Z}_q$ randomly and computing a public key $y = g^x$. Encryption of a plaintext $m \in G_q$ is defined by $E_y(m, r) = (g^r, y^r m)$, where $r \in \mathbb{Z}_q$ is chosen randomly. Decryption of a ciphertext (u, v) is defined by $D_x(u, v) = u^{-x} v$.

In class we sketched a reduction of the DDH problem to the problem of violating the CPA security of the El Gamal cryptosystem. This means that El Gamal is CPA-secure under the DDH assumption. You task is to write down this proof in detail.

Definition 1 (DDH problem). An algorithm $\mathcal{A}$ (δ, T)-solves the DDH problem in G_q if it runs in time T and

$$\left|\Pr[\mathcal{A}(g^a, g^b, g^{ab}) = 1] - \Pr[\mathcal{A}(g^a, g^b, g^c) = 1]\right| \geq \delta \ ,$$

where $a, b, c \in \mathbb{Z}_q$ are uniformly and independently distributed.

Definition 2 (CPA experiment). The CPA experiment $\mathsf{Exp}^b_{\mathcal{A}}$ for the El Gamal cryptosystem over G_q and an adversary $\mathcal{A}$ is defined as follows.

1. Generate a random El Gamal key pair (y, x).
2. Let the adversary choose plaintexts and output its state $(m_0, m_1, s) = \mathcal{A}(y)$.
3. Choose $r \in \mathbb{Z}_q$ and output $\mathcal{A}(s, E_y(m_b, r))$.

Definition 3 (CPA insecurity). An adversary $\mathcal{A}$ (δ, T)-violates the chosen plaintext attack security of El Gamal if it runs in time T and $\left|\Pr[\mathsf{Exp}^0_{\mathcal{A}} = 1] - \Pr[\mathsf{Exp}^1_{\mathcal{A}} = 1]\right| \geq \delta$.

Task 4.1 (2T). Explicitly define a hybrid experiment $\mathsf{Exp}^*_{\mathcal{A}}$ where a randomly chosen plaintext in G_q is encrypted instead of one of the two plaintexts m_0 and m_1 chosen by the adversary.

Task 4.2 (3T). Use a hybrid argument to show that if $\mathcal{A}$ (δ, T)-violates the chosen plaintext attack security of El Gamal, then there exists a $b \in \{0, 1\}$ such that $\left|\Pr[\mathsf{Exp}^b_{\mathcal{A}} = 1] - \Pr[\mathsf{Exp}^*_{\mathcal{A}} = 1]\right|$.

Task 4.3 (5T). Then use this to show that if $\mathcal{A}$ (δ, T)-violates the chosen plaintext attack security of El Gamal, then there exists an algorithm $\mathcal{A}$ that (δ', T')-solves the DDH problem in G_q, where $\delta' \geq \delta/2$ and $T' \leq 2T$. (You may assume that any trivial book keeping operations as well as a constant number of group operations can be done in time T.)

Problem 5 (Pseudo-random Generators). Consider two distinct functions f_1 and f_2 such that on input $x \in \{0, 1\}^n$ give outputs in $\{0, 1\}^{4n}$, i.e., they are expanding their inputs by a factor of 4. You know that at least one of the two functions is a pseudo-random generator, but not which one. Your goal is to construct a single pseudo-random generator under this assumption.

Task 5.1 (2T). Prove that $f(x) = f_1(x) \oplus f_2(x)$ is not necessarily a pseudo random generator, i.e., define distinct f_1 and f_2 (of which at least one is a pseudo-random generator) such that it is not and explain why. (You may assume that g is a pseudo-random generator that expands its input by a factor of 4 and use it to define suitable f_1 and f_2 as a counter example.)

Task 5.2 (4T). Prove that $f(x) = f_1(x_1) \oplus f_2(x_2)$, where x_1 and x_2 denote the first and second half of x is a pseudo-random generator with expansion 2, i.e., prove that if there is an adversary that violates the definition of a PRG for your function f, then there exists an adversary that violates it for f_1 and (a possibly different adversary) that violates it for f_2.

Problem 6 (Discrete Logarithms). Consider the Mersenne prime $p = 2^{107} - 1$ and an integer $0 < g < p$, namely $g = 44301866130425186081415332943872$. Interpret[1] your email address as an integer y as follows: (1) convert each character to its byte value, (2) concatenate the bytes, and (3) reduce modulo p.

Task 6.1 (1T). Compute $\log_g y$ in the additive group $\mathbb{Z}_p$ and explain your method.

Task 6.2 (2T). Explain how to efficiently determine the order of an element in the multiplicative group $\mathbb{Z}_p^*$, and compute it for your value y.

Problem 7 (Signature Schemes). Read about Lamport's one-time signatures that first computes a digest of a message and then signs the digest using Lamport's idea. Assume that the hash function is a random oracle with suitable range.

Task 7.1 (1T). Define the key generation, signature, and verification algorithms.

Task 7.2 (2T). How many signatures computed using distinct messages, but the same secret key, suffices to recover the complete secret key? Introduce suitable notation, restate the question mathematically, describe your attack, and prove that it works with probability at least $1/2$.

Problem 8 (Nitty Gritty Details). Let $p = kq+1$, where p is a 3248-bit prime, q is a 256-bit prime, and k factors into distinct primes of bitsize at most 32. Suppose that G_q is the unique subgroup of $\mathbb{Z}_p^*$ of size q generated by g.

You are given an El Gamal public key $y = g^x \bmod p$, where $x \in \mathbb{Z}_q$ is randomly chosen. Then you can ask me to decrypt any El Gamal ciphertext $(u, v) \in \mathbb{Z}_p^* \times \mathbb{Z}_p^*$, i.e., I will return $u^{-x}v \bmod p$ to you.

Task 8.1 (3T). Explain how to recover my secret key x.

Task 8.2 (1T). Can you help me avoid your attack?

Problem 9 (Zero Knowledge Proof of Knowledge). Let G and H be groups of prime order q and let $\phi : G \to H$ be a homomorphism.

Task 9.1 (2T). Describe a generalization of Schnorrs protocol for knowledge of discrete logarithms to the relation

$$\mathcal{R} = \big\{(x, w) \in G \times H \mid \phi(w) = x\big\}$$

that: (1) has perfect completeness, (2) is special sound, and (3) is *honest* verifier zero knowledge.

Task 9.2 (1T). Prove that it has perfect completeness.

Task 9.3 (1T). Prove that it is special sound.

Task 9.4 (1T). Prove that it is honest verifier zero knowledge.

---

[1] We do not really care about how you do this as long as you do something that gives you a personal unpredictible non-zero integer.