ETH Zürich, D-INFK
Spring 2018

Prof. Ueli Maurer
Christian Badertscher
Fabio Banfi

Cryptography Foundations

Exercise 9

9.1 Random Self-Reducibility of the Computational Diffie-Hellman Problem

Goal*: We consider the computational Diffie-Hellman problem as an example of random-self reducible problems.*

Prove that the CDH problem is random self-reducible.

9.2 Cloning the MAC-forgery Game

Goal*: The MAC-forgery game as presented in the lecture notes is not clonable. This task explores the reason and investigates a weaker variant of the game which can be cloned.*

A MAC for message space $\mathcal{M}$, key space $\mathcal{K}$, and tag space $\mathcal{T}$ is a function $f : \mathcal{M} \times \mathcal{K} \to \mathcal{T}$. The security of a MAC can be defined by a game $\mathbf{G}$ that allows the adversary to obtain valid MACs for chosen messages, and finally takes as input a pair (m, t) such that m has not been queried before. The game is won if the pair (m, t) constitutes a valid message/MAC-pair. We strengthen this definition such that the adversary has multiple attempts to forge; the game is won if at least one such attempt is successful.

a) Show that the "straightforward" system $\mathbf{K}$ that emulates q copies of the MAC-forgery game by simply forwarding the inputs and outputs does not achieve cloning, even if the MAC-forgery game allows multiple attempts to forge.

In a *fixed-target MAC-forgery game* the message for which the adversary has to forge a MAC is fixed in the beginning by the game system $\mathbf{G}_{\mathsf{fix}}$. More detailed, the game $\mathbf{G}_{\mathsf{fix}}$ can be described as:

- Generate the key $k \in \mathcal{K}$ uniformly at random and choose the target message $\hat{m} \in \mathcal{M}$ according to some distribution.[1] Output $\hat{m}$ at the right interface.
- On input a message $m \in \mathcal{M}$ at the right interface, if $m = \hat{m}$, then answer with $\perp$. Otherwise, compute $t = f(m, k)$ and answer with t.
- On input a MAC $\hat{t} \in \mathcal{T}$ at the right interface, set the output on the left interface to 1 if $\hat{t} = f(\hat{m}, k)$.

b) Show that the fixed-target MAC-forgery game with multiple verification queries is clonable.

[1] The distribution can be seen as a parameter of the game. The clonability holds independently of this distribution.

9.3 Performance Amplification Revisited

Goal*: While we have seen that one cannot generally amplify the success probability of a winner, we prove that this is possible if we make an additional assumption on the winner and the game.*

Let $\mathbf{G}$ be a probabilistic game that is 2-clonable by $\mathbf{K}$. For a winner $\mathbf{W}$, define the random variable $X_{\mathbf{W}} := \Pr^{\mathbf{W}}\big[\omega(\mathbf{W}, \mathbf{G}) = 1\big]$. Find a reduction π such that for all winners $\mathbf{W}$ with $\mathbb{E}\big[X_{\mathbf{W}}^2\big] < \mathbb{E}\big[X_{\mathbf{W}}\big]$,

$$\overline{\mathbf{G}}(\mathbf{W}) \;<\; \overline{\mathbf{G}}\;\pi(\mathbf{W})$$

and prove this.

9.4 Properties of the Distinguishing Advantage

Goal*: Recall the notion of a pseudo-metric and prove a related lemma of the lecture notes.*

Prove Lemma 4.7 from the lecture notes, i.e., show that for any $\mathcal{D}$ that is closed under complementing the output bit, $\Delta^{\mathcal{D}}$ is a pseudo-metric.

The following task considers the reading assignment. The reading assignment is considered lecture material. Note that on Wednesday (2.5.2018) no new lecture material is introduced.

9.5 Abstract Models of Computation

Goal*: This task is to improve your understanding of the type of results presented in the reading assignment and to apply one of the main theorems.*

Consider the following problem: Given the group $\{0,1\}^\ell$ with the bit-wise XOR, the goal is to extract an unknown value $x \in \{0,1\}^\ell$. The allowed operations are the group operation, the insertion of constant values $a \in \{0,1\}^\ell$, and checks for equality of two values.

a) Formalize the abstract model of computation for the above problem following the exposition in the reading assignment.

b) Provide a (non-trivial) algorithm that solves the above problem. How many operations and how many relation queries does the algorithm perform?

Hint: You might get some inspiration on which approach to follow by reading through the concrete algorithms described in the reading assignment.

c) Would a total order relation on any representation, as explained in the beginning of Section 4.7 of the reading assignment, help to improve your algorithm from subtask **b)**?

Hint: Such an arbitrary total ordering on the representation does not correspond to any property of the values.

d) Which Theorem of the reading assignment directly gives you a (non-trivial) lower bound on the number of operations to solve this problem (i.e., to achieve a constant success probability)?

Discussion of solutions:

Wednesday, 2.5.2018 (Tasks 9.1 and 9.2) **instead of the lecture**

(Note that we cancel the exercise sessions on 30.4.2018 and 1.5.2018)

Monday/Tuesday: 7/8.5.2018 (Tasks 9.3, 9.4, 9.5)