ETH Zürich, D-INFK
Spring 2018

Prof. Ueli Maurer
Christian Badertscher
Fabio Banfi

# Cryptography Foundations

# Solution Exercise 10

## 10.1 Hardness Amplification for Many Instances

**a)** Consider a function $\mu\colon \mathcal{S}_1 \times \ldots \times \mathcal{S}_k \to [0,1]$ and independent random variables $S_1, \ldots, S_k$ where $S_i$ has range $\mathcal{S}_i$. Abusing notation, we write for this exercise $\mathcal{S}^k := \mathcal{S}_1 \times \ldots \times \mathcal{S}_k$ and $\mathcal{S}^{k\setminus i} := \mathcal{S}_1 \times \ldots \times \mathcal{S}_{i-1} \times \mathcal{S}_{i+1} \times \ldots \times \mathcal{S}_k$. Similarly, we write for variables $s^k := (s_1, \ldots, s_k)$ as well as $s^{k\setminus i} := (s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_k)$. Moreover, we define $\mu_i(s_i)$ as the average of $\mu$ when $S_i = s_i$, i.e.,

$$\begin{aligned}\mu_i(s_i) &= \mathbb{E}_{S^{k\setminus i}}[\mu(S_1, \ldots, S_{i-1}, s_i, S_{i+1}, \ldots, S_k)] \\ &= \sum_{s^{k\setminus i} \in \mathcal{S}^{k\setminus i}} \left( \mu(s^k) \prod_{j \in \{1,\ldots,k\}\setminus\{i\}} \mathsf{P}_{S_j}(s_j) \right).\end{aligned}$$

Lemma 4.13 now becomes

$$\mathbb{E}_{S^k}[\mu(S^k)] \;\leq\; \prod_{i=1}^{k} \Pr^{S_i}\left[\mu_i(S_i) \geq \epsilon\right] + k\epsilon.$$

Let for $i \in \{1, \ldots, k\}$, $\mathcal{S}_i' = \{s \in \mathcal{S}_i \,|\, \mu_i(s) \geq \epsilon\}$ and $\mathcal{S}_i'' = \mathcal{S}_i \setminus \mathcal{S}_i'$.
Using

$$\mathcal{S}^k = \left(\mathcal{S}'\right)^k \cup \bigcup_{j=1}^{k} \left(\mathcal{S}_1 \times \ldots \times \mathcal{S}_{j-1} \times \mathcal{S}_j'' \times \mathcal{S}_{j+1} \times \ldots \times S_k\right),$$

we can bound

$$\begin{aligned}\mathbb{E}_{S^k}[\mu(S^k)] &= \sum_{s^k \in \mathcal{S}^k} \left( \mu(s^k) \cdot \prod_{i=1}^{k} \mathsf{P}_{S_i}(s_i) \right) \\ &\leq \sum_{s^k \in (\mathcal{S}')^k} \left( \mu(s^k) \cdot \prod_{i=1}^{k} \mathsf{P}_{S_i}(s_i) \right) + \sum_{j=1}^{k} \left( \sum_{(s_j, s^{k\setminus j}) \in \mathcal{S}_j'' \times \mathcal{S}^{k\setminus j}} \left( \mu(s^k) \cdot \prod_{i=1}^{k} \mathsf{P}_{S_i}(s_i) \right) \right).\end{aligned}$$

Since $\mu(s^k) \leq 1$ we have

$$\sum_{s^k \in (\mathcal{S}')^k} \left( \mu(s^k) \cdot \prod_{i=1}^{k} \mathsf{P}_{S_i}(s_i) \right) \;\leq\; \prod_{i=1}^{k} \Pr^{S_i}[S_i \in \mathcal{S}_i'] = \prod_{i=1}^{k} \Pr^{S_i}[\mu_i(S_i) \geq \epsilon].$$

Since $\mu_j(s_j) < \epsilon$ for $s_j \in \mathcal{S}''_j$, we further have for all $j \in \{1,\ldots,k\}$

$$\begin{aligned}\sum_{(s_j,s^{k\setminus j})\in\mathcal{S}''_j\times\mathcal{S}^{k\setminus j}} \left(\mu(s^k)\cdot\prod_{i=1}^{k}\mathsf{P}_{S_i}(s_i)\right) &= \sum_{s_j\in\mathcal{S}''_j}\mathsf{P}_{S_j}(s_j)\underbrace{\sum_{s^{k\setminus j}\in\mathcal{S}^{k\setminus j}}\left(\mu(s^k)\prod_{i\in\{1,\ldots,k\}\setminus\{j\}}\mathsf{P}_{S_i}(s_i)\right)}_{=\mu_j(s_j)}\\ &\le \Pr^{S_j}[S_j\in\mathcal{S}''_j]\cdot\epsilon\\ &\le \epsilon.\end{aligned}$$

This completes the proof.

**b)** Let $G_k^i$ be the generalization of the converter $\underline{G}$ as in Definition 4.14: At its left interface it provides access to the parallel composition of $k$ systems, where all but the $i$th are independent copies of $G$, which are emulated by $G_k^i$ and whose right outputs are ignored. The $i$th system is whatever is connected to $G_k^i$ on the right hand side.

Fix a winner $W$ for $G^{k^\wedge}$ and define the function $\mu(g^k) := \overline{[g_1,\ldots,g_k]^\wedge}(W)$, where $g_i$ is an instance of $G$. For $i = 1,\ldots,k$, let $G_i$ be independent random variables with the same distribution as $G$ and

$$\mu_i(g_i) := \overline{[G_1,\ldots,G_{i-1},g_i,G_{i+1},\ldots,G_k]^\wedge}(W).$$

We have

$$\mu_i(g_i) \;\le\; \overline{g_i}(WG_k^i)$$

since whenever $W$ wins $G^{k^\wedge}$, it in particular wins the $i$th instance.

Let for $i = 1,\ldots,k$ and $\epsilon = \delta'/k$,

$$A_i := \Pr^{G_i}[\mu_i(G_i)\ge\epsilon].$$

Lemma 4.19 implies

$$\overline{G^{k^\wedge}}(W) \;\le\; \prod_{i=1}^{k} A_i + k\epsilon.$$

Let $I$ be uniformly distributed over $\{1,\ldots,k\}$. As in the lecture notes (Proof of Theorem 4.12) we have

$$\overline{G}((WG_k^I)^qK) = \overline{G^{[q]^\vee}}((WG_k^I)^q) \ge \frac{A_1+\ldots+A_k}{k}\cdot\psi_q(\epsilon) \ge (A_1\cdot\ldots\cdot A_k)^{1/k} \;/\; (1+\delta)^{1/k},$$

which follows since (1) we choose $q$ large enough such that

$$\psi_q(\epsilon) \ge 1/(1+\delta)^{1/k}$$

holds, and (2) by the inequality of arithmetic and geometric means.

Hence, we have

$$\prod_{i=1}^{k} A_i \le (1+\delta)(\overline{G}((WG_k^I)^qK))^k.$$

This implies Theorem 4.14 for the reduction $\rho$ defined as

$$\rho(W) = \left(WG_k^I\right)^qK.$$

### 10.2 A Graph-Theoretic Result

Let $M$ be the adjacency matrix of $G$ and let $S$ and $T$ be independent and uniform random variables over $V$. Define $\mu\colon V \times V \to [0,1], (u,v) \mapsto M_{u,v}$, where $M_{u,v}$ is the entry in the $u$-th column and the $v$-th row of $M$. Note that $\mathbb{E}_{ST}[\mu(S,T)] = \alpha$. For $\mu_1\colon V \to [0,1], s \mapsto \mathbb{E}_T[\mu(s,T)]$ and $\mu_2\colon V \to [0,1], t \mapsto \mathbb{E}_S[\mu(S,t)]$, we have by Lemma 4.11,

$$\alpha = \mathbb{E}_{ST}[\mu(S,T)] \leq \Pr^S[\mu_1(S) \geq \epsilon] \cdot \Pr^T[\mu_2(T) \geq \epsilon] + 2\epsilon.$$

Since we have $\alpha \geq \beta^2 + 2\epsilon$, this implies that $\Pr^S[\mu_1(S) \geq \epsilon] \geq \beta$ or $\Pr^T[\mu_2(T) \geq \epsilon] \geq \beta$. This means that at least a $\beta$-fraction of the rows contain at least $\epsilon \cdot |V|$ ones, or at least a $\beta$-fraction of the columns contain at least $\epsilon \cdot |V|$ ones, which is equivalent to the claim.

### 10.3 Generic Reduction of the DL Problem to the CDH Problem

**a)** As suggested in the reading assignment (Section 4.8.7), we need to consider the extraction problem for the ring $\mathbb{Z}_p$ with additive operation $\oplus$ (addition modulo $p$) and multiplicative operation $\odot$ (multiplication modulo $p$). The model is described by a black box $\mathbf{B}$ with internal variables $V_i \in \mathbb{Z}_p$, such that $V_1$ is initialized to the secret value $x \in \mathbb{Z}_p$ chosen uniformly at random (this corresponds to the discrete logarithm of an element of the group $\mathbb{G}$). The allowed operations are the nullary operations $\pi_y$ for $y \in \mathbb{Z}_p$ that simply set the next ($j$-th) variable to $V_j := y$, the binary operation $\pi_\oplus(m,n)$ that sets $V_j := V_m \oplus V_n$, and the binary operation $\pi_\odot(m,n)$ that sets $V_j := V_m \odot V_n$. The only allowed query is the binary query $\sigma_=(m,n)$ that returns 1 if and only if $V_m = V_n$.

**b)** Let $h$ be a generator of $\mathbb{Z}_p^*$ that is assumed to be known. Then the reduction is specified by the three converters $\mathbf{C}_\Pi$, $\mathbf{C}_\Sigma$, and $\mathbf{C}_{\mathsf{out}}$.

- On input the operation $\pi_y$ (with $y \in \mathbb{Z}_{p-1}$) at the outside of $\mathbf{C}_\Pi$, the operation $\pi_{h^y}$ (with $h^y \in \mathbb{Z}_p^*$) is output at the inside of $\mathbf{C}_\Pi$. On input the operation $\pi_\oplus(m,n)$ at the outside of $\mathbf{C}_\Pi$, the operation $\pi_\odot(m,n)$ is output at the inside of $\mathbf{C}_\Pi$.
- On input the query $\sigma_=(m,n)$ at the outside of $\mathbf{C}_\Sigma$, the query $\sigma_=(m,n)$ is output at the inside of $\mathbf{C}_\Sigma$.
- On input the value $w \in \mathbb{Z}_{p-1}$ at the inside of $\mathbf{C}_{\mathsf{out}}$, the value $h^w \in \mathbb{Z}_p^*$ is output at the outside of $\mathbf{C}_{\mathsf{out}}$.

Brief justification of this reduction (recall the ideas of exercise 7.2): the reduction simply changes the internal representation of the values and inputs of $\mathcal{A}$ via an isomorphism $\phi_h : \mathbb{Z}_{p-1} \to \mathbb{Z}_p^*$, i.e., $\phi_h(y) = h^y$. Thanks to this homomorphism, we have $\phi(x+y) = h^x \odot h^y$, (which is to say that we represent the addition as a multiplication), and thanks to the bijective property, the equality queries are answered correctly. Hence, the algorithm $\mathcal{A}$ which solves the extraction problem for (any element of) the additive group $\mathbb{Z}_{p-1}$ can be used to compute the correct result for the extraction problem for the multiplicative group $\mathbb{Z}_p^*$.

**c)** After we have shown how to translate (efficiently) any algorithm $\mathcal{A}$ that solves the extraction problem for $\mathbb{Z}_{p-1}$ to an algorithm that solves the extraction problem for $\mathbb{Z}_p^*$, we invoke the results from the reading assignment (or exercise 3.3): the Pohlig-Hellman algorithm[1] described in Section 4.6.3 of the reading assignment (coupled with BSGS as indicated there) requires $O(\sqrt{q'}\log n)$ operations where $q'$ is the largest prime factor of $n := |\mathbb{Z}_{p-1}| = p-1$ and by $B$-smoothness we have $q' \leq B$. If $B$ is treated as a constant, we have $O(\log n)$, i.e., the runtime grows linear in the input size (which are elements of a group of size $n$).

[1] We assume that the factorization of $p-1$ is known.

This illustrates the basic idea behind the reduction mentioned in Section 4.8.7 of the reading assignment. The full result is found in the references given in Section 4.8.7 (which is not part of this course) and considers elliptic curves instead of the simpler objects $\mathbb{Z}_p^*$ to find a good reduction.