Solution

Cryptography – Homework 2

Discussed on Tuesday, 4th November, 2014.

For questions regarding the exercises, please send an email to schlund@in.tum.de or just drop by at room 03.11.055

Exercise 2.1 Negligible Functions

(a) Show that the following two definitions are equivalent:

i) $\varepsilon(n)$ is negligible if and only if $\forall a \in \mathbb{N} \exists N \in \mathbb{N} \forall n > N : \varepsilon(n) < n^{-a}$.

ii) $\varepsilon(n)$ is negligible if and only if $\forall$polynomials q $\exists N \in \mathbb{N} \forall n > N : \varepsilon(n) < \frac{1}{|q(n)|}$

(b) Let $\varepsilon_1 : \mathbb{N} \to \mathbb{R}^+, \varepsilon_2 : \mathbb{N} \to \mathbb{R}^+$ be negligible functions and let $p : \mathbb{N} \to \mathbb{R}^+$ be a polynomial in $\mathbb{R}$. Show that $f : \mathbb{N} \to \mathbb{R}^+$ with $f(n) = \varepsilon_1(n) + \varepsilon_2(n)$ and $g : \mathbb{N} \to \mathbb{R}^+$ with $g(n) = p(n) \cdot \varepsilon_1(n)$ are also negligible functions.

(c) Prove or disprove for each of the following functions whether they are negligible:

- $2^{-\log n^3}$
- $\frac{1}{n^{23} \cdot \log(n!)}$
- $3^{-\sqrt{n}}$

(d) If $\varepsilon : \mathbb{N} \to \mathbb{R}^+$ is negligible, then $f : \mathbb{N} \to \mathbb{R}^+$ with $f(n) := \varepsilon(\lceil \sqrt{n} \rceil)$ is also negligible.

Solution: A function $\varepsilon : \mathbb{N} \to \mathbb{R}^+$ is negligible iff for every polynomial $q(\cdot)$ there is an N s.t. $\forall n \geq N : \varepsilon(n) < 1/|q(n)|$. Let $q(\cdot)$ be an arbitrary polynomial.

(a) The second definition clearly implies the first one (n^a are just a special polynomials). Thus let us assume $\varepsilon(n)$ satisfies the first definition and show that it also satisfies the second one. To this end, fix some polynomial q. We can write q as $q(n) = a_K n^K + a_{K-1} n^{K-1} + \dots a_1 n + a_0$. Since ε is negligible (according to the first def.) we know that $\varepsilon(n) < n^{-(K+1)}$ for n larger than some $N \in \mathbb{N}$. Since $n^{-(K+1)} < 1/|q(n)|$ for n larger than some other $N' \in \mathbb{N}$ we obtain that $\varepsilon(n) < 1/|q(n)|$ for $n > \max(N, N')$.

(b)
- Let $N_1, N_2 \in \mathbb{N}$ be such that for $i \in \{1, 2\}$, $\forall n \geq N_i : \varepsilon_i(n) < 1/|2q(n)|$. Now choose $N = \max(N_1, N_2)$ and let $n \geq N$. Then
$$f(n) = \varepsilon_0(n) + \varepsilon_1(n) < \frac{1}{|2q(n)|} + \frac{1}{|2q(n)|} = \frac{1}{|q(n)|}.$$
- Let N_1 be such that $\forall n \geq N_1 : \varepsilon_1(n) < 1/|p(n) \cdot q(n)|$. Now choose $N = N_1$ and let $n \geq N$. Then
$$g(n) = p(n) \cdot \varepsilon_1(n) < p(n) \cdot \frac{1}{|p(n) \cdot q(n)|} = \frac{1}{|q(n)|}$$
(note that $p(n) > 0$).

(c)
- $2^{-\log n^3} = 2^{\log n^{-3}} = n^{-3} \geq n^{-3}$ for all n, hence the function is *not* negligible.
- From $n! \leq n^n$ it follows that $\log(n!) \leq n \log n \leq n^2$ and therefore
$$\frac{1}{n^{23} \cdot \log(n!)} \geq n^{-25}$$
which implies that the function is *not* negligible.
- For any $a \in \mathbb{N}$ it holds that
$$\lim_{n \to \infty} \frac{n^a}{3^{\sqrt{n}}} = 0.$$
Hence $3^{\sqrt{n}}$ is negligible.

(d) *(Remark: the following is a very detailed and lengthy writeup, much more detailed than you would be expected to provide – but maybe it is instructive ;))*

Let $\varepsilon(n)$ be negligible, i.e. $\forall a \in \mathbb{N} \exists N_a \in \mathbb{N} \forall n > N_a : \varepsilon(n) < n^{-a}$. We have to show that $\varepsilon(\lceil\sqrt{n}\rceil)$ is negligible.

To this end, let $b \in \mathbb{N}$ be arbitrary, it remains to show

$$\exists N_b \in \mathbb{N} \forall n > N_b : \varepsilon(\lceil\sqrt{n}\rceil) < n^{-b}.$$

For $a := 2b$ we know that there exists an N_a such that $\forall n > N_a : \varepsilon(n) < n^{-a}$, which means

$$\forall \lceil\sqrt{n}\rceil > N_a : \varepsilon(\lceil\sqrt{n}\rceil) < \lceil\sqrt{n}\rceil^{-2b}$$

Now recall that $x \leq \lceil x \rceil$ and hence $n > N_a^2$ implies $\lceil\sqrt{n}\rceil > N_a$. Therefore, if we choose $N_b := N_a^2$ we know that the following holds:

$$\forall n > N_b : \varepsilon(\lceil\sqrt{n}\rceil) < \frac{1}{\lceil\sqrt{n}\rceil^{2b}} \leq \frac{1}{\sqrt{n}^{2b}} = n^{-b}$$

which is what we wanted to show.

Exercise 2.2 Pseudorandom Generators

- Let $f\colon \{0,1\}^* \to \{0,1\}$ be any DPT-computable function.

  Show that $G_f(x) = x||f(x)$ is no PRG.
- Use a similar argument to show that also the follwing is not a PRG:

  Let $m \in \mathbb{N}$ and $a, c \in \mathbb{Z}_m$.
  - For simplicity, let $m = 2^n$ so that we may identify $\mathbb{Z}_m$ with $\{0,1\}^n$.

  For $x \in \{0,1\}^n$, let $f(x) = (a \cdot x + c) \bmod m$, and $G(x, 1^{n \cdot s}) = f(x)||f(f(x))||\ldots||f^s(x)$.

Solution: A possible $\mathcal{D}$ works as follows:

On input $y = y_1 \ldots y_n y_{n+1} \in \{0,1\}^{n+1}$, output 1 iff $y_{n+1} = f(y_1 \ldots y_n)$. Now:

(a) $\Pr_{b=1}\left[\mathsf{Win}^{\text{IndPRG}}_{n,G}(\mathcal{D})\right] = \Pr_{x \overset{u}{\in} \{0,1\}^n}[\mathcal{D}(G(x)) = 1] = 1$ by definition of G.

(b) $\Pr_{b=0}\left[\mathsf{Win}^{\text{IndPRG}}_{n,G}(\mathcal{D})\right] = \Pr_{y \overset{u}{\in} \{0,1\}^{n+1}}[\mathcal{D}(y) = 0] = 1/2$:

Given $y_1 \ldots y_n$ the value $f(y_1 \ldots y_n)$ is already fixed. As the last bit y_{n+1} is chosen uniformly and independent of the other bits, with prob. 1/2 we have $y_{n+1} = f(y_1 \ldots y_n)$.

(c) Together we obtain: $\Pr\left[\mathsf{Win}^{\text{IndPRG}}_{n,G}(\mathcal{D})\right] = 1/2(1 + 1/2) = 3/4$ which is non-negligibly better than 1/2.

The second question works identically (it can even be seen as a special case of the first one).

Exercise 2.3 Pseudorandom Generators II

Let G be a PRG of stretch $l(n) = 2n$.

(a) Show that there exists an *exponential time* distinguisher $\mathcal{D}$ with:

$$\left|\Pr_{x \overset{u}{\in} \{0,1\}^n}[\mathcal{D}(1^n, G(x)) = 1] - \Pr_{y \overset{u}{\in} \{0,1\}^{2n}}[\mathcal{D}(1^n, y) = 1]\right| \geq 1 - 2^{-n}$$

(b) Determine the success probability of the following $\mathcal{D}$:
- Input: $y \in \{0,1\}^{l(n)}$ and 1^n.
- Generate $x' \overset{u}{\in} \{0,1\}^n$.
- Compute $y' = G(x')$.
- Return 1 if $y = y'$; else return 0.

Solution:

(a) The exponential time distinguisher works as follows: on input x check if $x \in G(\{0,1\}^n)$ (e.g. by enumerating all images of G). If so answer $r = 1$ else answer $r = 0$. If $b = 0$ then $\mathcal{D}$ will loose with probability at most 2^{-n} (i.e. the probability that a truly random string from $\{0,1\}^{l(n)}$ appears in the image of G). If $b = 1$ it will win with probability 1.

(b) If $b = 1$, then $\Pr_{b=1}\left[\mathsf{Win}_{n,G}^{\text{IndPRG}}(\mathcal{D})\right] = \Pr_{x \overset{u}{\in} \{0,1\}^n}\left[\mathsf{Win}_{n,G}^{\text{IndPRG}}(\mathcal{D})\right] \geq \Pr_{x \overset{u}{\in} \{0,1\}^n}[x' = x] = 2^{-n}$.

If $b = 0$, then

$$\Pr_{b=0}\left[\mathsf{Win}_{n,G}^{\text{IndPRG}}(\mathcal{D})\right] = 1 - \Pr_{b=0}[\mathcal{D}(y) = 1] = 1 - \sum_{y' \in G(\{0,1\}^n)} \Pr[y = y', G(x') = y']$$

$$\overset{y,x' \text{ indep.}}{=} 1 - \sum_{y' \in G(\{0,1\}^n)} \underbrace{\Pr[y = y']}_{=2^{-2n}} \Pr[G(x') = y'] = 1 - 2^{-2n} \underbrace{\sum_{y' \in G(\{0,1\}^n)} \Pr[G(x') = y']}_{=1} = 1 - 2^{-2n}$$

Together we obtain: $\Pr\left[\mathsf{Win}_{n,G}^{\text{IndPRG}}(\mathcal{D})\right] \geq \frac{1}{2}2^{-n} + \frac{1}{2}(1 - 2^{-2n}) = 1/2 + \underbrace{\frac{1}{2}(2^{-n} - 2^{-2n})}_{>0}$

Conclusion: This $\mathcal{D}$ has always a negligible, but non-zero advantage in distinguishing the PRG from a truely random source. Hence, if we had required that every $\mathcal{D}$ has zero advantage, then no PRG (of stretch $l(n) \geq 2n$) could exist w.r.t. this definition.

Exercise 2.4 Stream cipher

We can use a PRG G of variable stretch to define a "G-steam cipher" ("variable prOTP"):

- $\mathcal{K} = \{0,1\}^n$, and $\mathcal{M}_n = \mathcal{C}_n = \{0,1\}^*$
- $\mathsf{Gen}(1^n) : k \overset{u}{\in} \{0,1\}^n$
- $\mathsf{Enc}_k(m) = G(k, 1^{|m|}) \oplus m$
- $\mathsf{Dec}_k(c) = G(k, 1^{|c|}) \oplus c$

Show that the G-stream cipher is comp. secret if G is a PRG of variable stretch.

Hint: Let $\mathcal{A}$ be a PPT-attack on $\mathcal{E}$ in the game IndEd. Denote by $T_\mathcal{A}$ the running time of $\mathcal{A}$. Construct again from $\mathcal{A}$ a distinguisher $\mathcal{D}$ for $G_{T_\mathcal{A}(\cdot)}$. In order to simulate the encryption algorithm, make use of the prefix property of G.

Remark: By treating multiple messages as a single "long" message, the G-stream cipher can easily be transformed into a stateful ES which has *indistinguishable multiple encryptions in the presence of an eavesdropper* (see the exercise on the "q-time pad").

One disadvantage of stateful ES is that Alice and Bob have to make sure that their instances of Enc_k (and Dec_k) are indeed in the same state. E.g. if they want to exchange messages in both directions, Bob needs to know "how far Alice has stretched the secret key k" so that he can use a fresh part of the output of $G(k, 1^s)$.

Discuss possible solutions for synchronizing the state.

Solution: We let AliceBob play the IndPRG-game against Eve who in turn plays the IndEd-game against $\mathcal{A}$. We show: If Eve can win the eavesdropping-game, then she can also win the distinguishing-game (with non-negligible advantage).

AliceBob	Eve ($\mathcal{D}$)
$b \overset{u}{\in} 0,1$ if $b = 0$ send $y \overset{u}{\in} \{0,1\}^T$ else $k \overset{u}{\in} \{0,1\}^n$ and send $y = G(k, 1^T)$	
	run $\mathcal{A}(1^n)$ and receive m_0, m_1 from $\mathcal{A}$ $b' \overset{u}{\in} \{0,1\}$ $c := m_{b'} \oplus y[1 \dots \lvert m_{b'}\rvert]$ send c to $\mathcal{A}$ receive guess r' from $\mathcal{A}$ if $b' = r'$ then output 1 else output 0

Case $b = 0$: $\mathcal{D}$ wins the IndPRG-game iff $b' \neq r'$ iff $\mathcal{A}$ looses the IndEd-game against the one-time-pad. The probability for this is exactly $1/2$.

Case $b = 1$: $\mathcal{D}$ wins the IndPRG-game iff $b' = r'$ iff $\mathcal{A}$ wins the IndEd-game against the "variable prOTP". The probability for this to happen is by definition $\Pr\left[\mathsf{Win}^{\text{IndEd}}(\mathcal{A})\right]$.

Together we have: $\Pr\left[\mathsf{Win}^{\text{IndPRG}}(\mathcal{D})\right] = 1/2 \cdot 1/2 + 1/2\Pr\left[\mathsf{Win}^{\text{IndEd}}(\mathcal{A})\right]$ and thus

$$\Pr\left[\mathsf{Win}^{\text{IndPRG}}(\mathcal{D})\right] - 1/2 = 1/2 \cdot (\Pr\left[\mathsf{Win}^{\text{IndEd}}(\mathcal{A})\right] - 1/2)$$

So if the advantage of winning the IndPRG-game is negligible then the G-stream cipher is computationally secret.

Remark: We need to know an upper bound $T = T_{\mathcal{A}}$ on the length of the messages m_0, m_1 to be sure to get a one-time-pad encryption (this is a small non-constructive part of the attack).

Possible solutions to synchronize the state:

- Alice and Bob use two keys $k_{A\to B}, k_{B\to A}$: Alice uses $k_{A\to B}$ to encrypt messages sent to Bob.
- Alice and Bob alternate and exchange all the time messages of fixed length (e.g. send 0^n if nothing to tell at the moment).