1. (12 points) Write a program that increments a counter $2^{24}, 2^{25}, 2^{26}, \ldots, 2^{33}$ times, and measure how many seconds your program takes to run in each case (list the machine specifications and the programming language you have used). Estimate how many years your program would take to increment a counter $2^{64}$ or $2^{128}$ times.

2. Recall that when defining EAV-security (as well as other notions of security), we required that the length of the messages output by the adversary satisfy $|m_0| = |m_1|$. The purpose of this problem is to demonstrate that we cannot remove this constraint for any encryption scheme that allows arbitrary length messages (that is, intuitively, encryptions always leak some information about the message length).

For the rest of the problem, consider only encryption schemes defined over the message space $\mathcal{M} = \{0, 1\}^*$ (arbitrary length messages).

(a) (4 points) Show that for any encryption scheme there is a fixed polynomial $q$ such that the length of the encryption of a 1-bit message is always at most $q(n)$, where $n$ is the security parameter.

(b) (6 points) Show that there is a message of polynomial (in $n$) length, which has no encryption of length $\leq q(n)$.

(c) (6 points) Deduce that EAV-security cannot be satisfied if in the definition of the experiment Priv_{A,\Pi}^{ev}(n, b) we remove the restriction on the adversary to output equal length messages.

3. (12 points) Prove that definitions 3.9 and definition 3.8 are equivalent (Definition 3.9 is the one we used for EAV-security, while 3.8 is the one where the challenge ciphertext is chosen as an encryption of $m_b$ for a random $b \in \{0, 1\}$, and the adversary needs to guess $b$ with probability non-negligibly better than $1/2$).

4. (30 points) Let $G : \{0, 1\}^* \rightarrow \{0, 1\}^*$ be a PRG. For each of the following constructions, prove whether it is necessarily a PRG or not. (The first three constructions start by parsing the input as two equal-length parts; $\overline{s}$ is the bitwise complement of $s$)

(a) $G_1(s, t) = \overline{s} \parallel G(t)$

(b) $G_2(s, t) = \overline{s} \parallel G(s, t)$

(c) $G_3(s, t) = s \parallel G(s, t) \parallel G(\overline{s}, t)$

(d) $G_c(s) = G(s) \parallel G(\overline{s})$

(e) $G_z(s) = G(s) \oplus z(|s|)$ (where $z()$ is a public and efficiently computable function of the security parameter $|s|$, and has the appropriate output length).

5. Extra Credit: For any of the constructions in problem 4 where you answered no (i.e., the construction is not necessarily a PRG), determine whether the construction could sometimes work (namely, if any PRG exists, can you come up with a PRG $G$ such that the construction built on top of $G$ is also a PRG?) Prove your answers.

---

1This can be addressed by padding the messages, but still an upper bound on the length is leaked.