CSE3400 HW1

Isaac Piegat

September 10, 2024

1 Problem 1

I used python on a Windows 11 octa-core AMD Ryzen 9 8945HS.

1.1 Part 1

- Time for 2^{10}: 0.0 seconds
- Time for 2^{20}: 0.07856202125549316 seconds
- Time for 2^{30}: 82.6774411201477 seconds

Estimated time for Part 1 (2^{480}): 89208.15517592432 seconds

1.2 Part 2

- Time for 2^{10}: 0.0 seconds
- Time for 2^{20}: 0.13591599464416504 seconds
- Time for 2^{30}: 139.3803141117096 seconds

Estimated time for Part 2 (2^{480}): 150382.65722608566 seconds

2 Problem 2:

Let $G_0 : \{0,1\}^n \to \{0,1\}^{2n}$ and $G_1 : \{0,1\}^{n/2} \to \{0,1\}^n$ be PRGs.

2.1 Part 1: $G_2(s \parallel t) = G_0(s) \parallel (t \oplus 1^n)$

The construction is NOT a PRG because of the vulnerability of $t \oplus 1^n$. The xor operation only flips the bits of t, so the attacker would only need to guess t. As t is a random string with length n, there would be 2^n possible combinations, thus the attacker would have a $1/2^n$ probability of success.

2.2 Part 2: $G_3(s \parallel z) = G_0(s \oplus z)$

The construction IS a PRG because of its unpredictability. The attacker, even if they knew s and z, would have no way to find the string produced by the pseudorandom operation of G_0. The probability of guessing the binary string of length $2n$ from G_3 would be $1/2^{2n}$.

2.3 Part 3: $G_4(s \parallel z) = LH(G_0(s) \oplus G_1(RH(z)))$, where LH is the left half of the input string, and RH is the right half of the input string

The construction IS a PRG because of its unpredictability. G_0 is a random binary string of length $2n$, and G_1 is a random binary string of length n. ($G_0 \oplus G_1$) would simply result in another random string of length n, thus guessing would have a probability of $1/2^{2n}$.

2.4 Part 4: $G_5(s) = (G_0(s) mod 2) \parallel G_0(s)$, where mod is the modulus operation.

The construction IS NOT a PRG because it is a recognizable pattern. The first and second half of the output string with length $4n$ would mirror each other because the xor operation does nothing to G_0 since neither 0 or 1 have a remainder when divided by two, effectively not changing the string. As the attacker would only need to guess the first half of the string, the probability would be $1/2^{2n}$.

3 Problem 3:

3.1 Part 1:

1. none
2. none