# Problem Set 3

Alex Miller

October 21, 2021

## 1

Let $l, n$ be positive integers, $n, l \geq 1$. For any distinguisher $D : \{0,1\}^l \to \{0,1\}$:

$Adv_G^{prg}(D) = |Pr_K[D(G(k)) = 1] - Pr_U[D(u) = 1]|$

where $k$ and $u$ are uniform random variables on $K = \{0,1\}^n, U = \{0,1\}^l$, respectively.

Say that $Adv_G^{prg}(D) = 1$ for some $G : \{0,1\}^n \leftarrow \{0,1\}^l$. There are two cases in which this is true:

- Case 1. $Pr_K[D(G(k)) = 1] = 1$ and $Pr_U[D(u) = 1] = 0$. However, this can never be true.
    1. $Pr_K[D(G(k)) = 1] = 1 \to \forall k \in K, D(G(k)) = 1$
    2. Moreover, since the set $\{G(k) : k \in K\} \subseteq U$ then $\exists s \in U s.t. D(s) = 1$
    3. However, by our second assumption $Pr_U[D(u) = 1] = 0 \to \forall u' in U, D(u) = 0$
    4. Our assumptions are therefore incompatible. This case never happens.
- Case 2. $Pr_K[D(G(k)) = 1] = 0$ and $Pr_U[D(u) = 1] = 1$. Again this is never true.
    1. $Pr_K[D(G(k)) = 1] = 0 \to \forall k' \in K, D(G(k)) = 0$
    2. Moreover, since the set $\{G(k) : k \in K\} \subseteq U$ then $\exists s \in U s.t. D(s) = 0$
    3. However, by our second assumption $Pr_U[D(u) = 1] = 1 \to \forall u' in U, D(u) = 1$
    4. Our assumptions are therefore incompatible. This case never happens.
- Since either case for which $Adv_G^{prg}(D) = 1$ is impossible, $Adv_G^{prg}(D) \neq 1$
- **QED**

# 2

Define the following $D : \{0,1\}^l \to \{0,1\}$ where $l \geq 2, lmod2 == 0$:

**Algorithm 1:** D(s)

1 *//Split our input string s in half*
2 $a \leftarrow s[: l/2], b \leftarrow s[l/2 :]$
3 return ($a$ XOR $b == \{1\}^{l/2}$) ? 1 : 0

- $Pr_K[D(G(k)) = 1] = 1$, where $k$ is a uniform random variable over $K = \{0,1\}^n, n \geq 1$
  - Consider output any $s \in \{G(k), k \in K\}$
  - $!\exists k \in \{0,1\}^n$, s.t. $s = k||\neg k$. In our definition of $D(s)$ we split $s$ into two strings $s.t. a = k, b = \neg k$
  - The result of $a \oplus b$ is equivalent to $k \oplus \neg k$
  - Moreover, $\forall k \in \{0,1\}^n, k$ XOR $\neg k = \{1\}^n$
  - Therefore $\forall s \in \{G(k), k \in K\}, D(s) = 1$
  - Therefore $D(s)$ returns 1 for any number generated by $G$
- $Pr_U[D(u) = 1] = \frac{1}{2^n}$, where $u$ is a uniform random variable over $u = \{0,1\}^{2*n}, n \geq 1$
  - We already showed that $\forall s \in \{G(k), k \in K\}, D(s) = 1$
  - Considering the definition of $G$, we can say then that $\forall (s = a||b) \in \{0,1\}^l$, if $a = \neg b, D(s) = 1$
  - We must show the opposite, namely that $\forall (s = a||b) \in \{0,1\}^l$, if $a \neq \neg b$, then $D(s) = 0$:
    - Consider any $(s = a||b) \in \{0,1\}^l$ s.t. $a \neq \neg b$
    - $a$ XOR $b \neq 1^n$ ($a$ and $b$ are not complements of each other)
    - Therefore $D(s) = 0$
  - The count of elements in $\{s : s \in U, D(s) = 1\} = \{s : (s = a||b) \in \{0,1\}^l, a = \neg b\} = \{G(k) : k \in K\} = 2^n$
  - The size of $U$ is $2^{2n}$
  - Since $u$ is a uniform random variable over $U$, $Pr_U[D(u) = 1] = \frac{2^n}{2^{2n}} = \frac{1}{2^n}$
- $Adv_G^{prg}(D) = |Pr_K[D(G(k)) = 1] - Pr_U[D(u) = 1]| = |1 - \frac{1}{2^n}| = \frac{2^n - 1}{2^n}$
- $Lim_{n \to \infty} Adv_G^{prg}(D) = 1$. Very large!

# 3

Note: Let $G_{n,l}(k \in \{0,1\}^n)$ denote some LSFR s.t. $G_{n,l} : \{0,1\}^n \to \{0,1\}^l$. For our purposes $\forall l, !\exists n, G_{n,l}$ uses a set of taps $V_n \subseteq \{0,1,...,n-1\}$; all LSFRs use the same set of taps as all other possible LSFRs that take in the same size input. Furthermore, $\exists n \geq 1, k \in \{0,1\}^n$, let $V_n(k) : \{0,1\}^n \to \{0,1\}$ describe the process of XORing all bits indexed by $V_n$ in $k$ (this is how stream ciphers generate bits, given a register state of $k$)

We show that $\forall l, n \geq 1, \forall k, k' \in K = \{0,1\}^n, G(k) \oplus G(k') = G(k \oplus k')$

*Proof by strong induction*

Base Case: $\forall n \geq l \geq 1, \forall k, k' \in K = \{0,1\}^n, G_{n,l}(k) \oplus G_{n,l}(k') = G_{n,l}(k \oplus k')$

- $\forall n \geq l \geq 1, \forall k \in K$, the first $l$ bits generated by $G_{n,l}(k)$ are equivalent to that of $k$.
- Therefore, $\forall n \geq l \geq 1, \forall k, k' \in K, G_{n,l}(k) \oplus G_{n,l}(k') = G_{n,l}(k \oplus k')$ because, in this case, $G_{n,l}(k) \oplus G_{n,l}(k') = G_{n,l}(k \oplus k')$ follows from $k \oplus k' = k \oplus k'$, which is always true.
- **QED**

Inductive Step: Consider any positive integer $l'$ such that $l' = l+1 > l > n$. We assume $\forall l > n \geq 1, \forall k, k' \in K = \{0,1\}^n, G_{n,l}(k) \oplus G_{n,l}(k') = G_{n,l}(k \oplus k')$. We show that, if this is true, $\forall k, k' \in K = \{0,1\}^n, G_{n,l'}(k) \oplus G_{n,l'}(k') = G_{n,l'}(k \oplus k')$.

- $\forall k, k' \in K = \{0,1\}^n$, consider the last $n$ of the first $l$ bits produced by the LSFRs from our inductive assumption. Let these sets of bits be known as:
  1. $A$ from $G_{n,l}(k)$
  2. $B$ from $G_{n,l}(k')$
  3. $C$ from $G_{n,l}(k \oplus k')$
- Now consider the $l' = l + 1$-th bit produced by the LSFRs from our inductive step. Let these bits be known as:
  1. $a'$ from $G_{n,l'}(k)$
  2. $b'$ from $G_{n,l'}(k')$
  3. $c'$ from $G_{n,l'}(k \oplus k')$
- We know a couple things!
  - For one, we know that $A, B$, and $C$ comprise states under which the LSFRs $G_{n,l'}(k)$, $G_{n,l'}(k')$, and $G_{n,l'}(k \oplus k')$ encoded the bits $a', b'$,and $c'$, respectively. (By the definition of an LSFR)
  - We also know that $A \oplus B = C$ (Inductive Assumption)

  - Furthermore, by our notation, we can say:
    * $a' = V_n(A)$
    * $b' = V_n(B)$
    * $c' = V_n(C)$

- We can use this to say that $a' \oplus b' = c'$
  - $V_n(A) \oplus V_n(B) = (a_{v_1} \oplus a_{v_2} \oplus ... \oplus av_v) \oplus (b_{v_1} \oplus b_{v_2} \oplus ... \oplus bv_v)$
  - $V_n(C) = V_n(A \oplus B) = ((a_{v_1} \oplus b_{v_1}) \oplus (a_{v_2} \oplus b_{v_2}) \oplus ... \oplus (a_{v_v} \oplus b_{v_v}))$
  - $V_n(A) \oplus V_n(B) = V_n(C)$(!) (XOR is associative)
- Observe that, because $a', b'$, and $c'$ are the $l'$-th bits produced by $G_{n,l'}(k), G_{n,l'}(k')$, $G_{n,l'}(k \oplus k')$, and $G_{n,l}(k) \oplus G_{n,l}(k') = G_{n,l}(k \oplus k')$, we can say that $G_{n,l'}(k) \oplus G_{n,l'}(k') = G_{n,l'}(k \oplus k')$
- Therefore if $\forall l > n \geq 1, \forall k, k' \in K = \{0,1\}^n, G_{n,l}(k) \oplus G_{n,l}(k') = G_{n,l}(k \oplus k')$, then $G_{n,l'}(k) \oplus G_{n,l'}(k') = G_{n,l'}(k \oplus k')$
- **QED**

Therefore, we've shown that for all cases of $\forall l, n \geq 1, \forall k, k' \in K = \{0,1\}^n, G(k) \oplus G(k') = G(k \oplus k')$ - our base case lets us induce on all combinations of $n$ and $l$ that meets its condition.
**QED**

# 4

Note: subscript notation on bit strings indicates an indexed bit

## 4.1

Bits are 0-indexed, and assume that $l > n = 256$
Let $V_1(s)$ represent the function by which $G_1$ generates a new bit given a register state $s$, where $s$ is a bit string $\in \{0,1\}^n$. $V_1(s) = s_{18} \wedge s_{193} \wedge s_{221}$
We can define $D : \{0,1\}^l \to \{0,1\}$ as:

**Algorithm 2:** $D_1(s)$

```
if V_1(s[: n]) == s[n] then
    return 1
end
return 0
```

Now let's calculate $Adv^{prg}_{G_1}(D_1) = |Pr_K[D_1(G_1(k)) = 1] - Pr_U[D_1(u) = 1]|$, where $k$ and $u$ are uniform random variables on $\{0,1\}^n$ and $\{0,1\}^l$ respectively
First, we evaluate $Pr_K[D_1(G_1(k)) = 1]$

- For any execution of $G_1$, the $n+1$th bit output by $G_1$ was generated using the register state initialized by the key $k$ ($G_1$ is an LSFR)

- This state is copied into the first $n$ bits generated by $G_1$. Call this sub string $k'$ ($G_1$ is an LSFR)
- We observe that $k = k'$ and therefore $V_1(k) = V_1(k')$;
- Moreover, since $V_1$ implements the method by which $G_1$ generates new bits, $\forall s \in \{G_1(k) : k \in \{0,1\}^n\}, k' = s[:n], V_1(k') = s_n$
- $\forall s \in \{0,1\}^l, k' = s[:n], D_1(s)$ operates by evaluating $V_1(k)' = s_n$ and returning 1 if the equality holds, and 0 otherwise
- Therefore $\forall s \in \{G_1(k) : k \in \{0,1\}^n\}, D_1(s) = 1$
- This means that $Pr_K[D_1(G_1(k)) = 1] = 1$
- **QED**

Next, we need to evaluate $Pr_U[D_1(u) = 1]$

- We already showed that $D_1$ returns 1 for all strings produced by $G_1$, but $\{G_1(k) : k \in \{0,1\}^n\} \subseteq U$, so we need to account for false positives returned by $D_1$
- Let $E$ describe the event $u_n = 0$, where $u$ is a uniformly random string on $U$. $Pr_U[E] = Pr_U[\neg E] = \frac{1}{2}$
- Let $F$ describe the event that $V_1(u[:n]) = 0$. Since the bits tapped by $V_1$ are uniformly random, we can count results of a truth table to arrive at a probability for $F$. Doing so, we get $Pr_U[F] = 0.875$ and $Pr_U[\neg F] = 0.125$
- Therefore $Pr_U[D_1(u) = 1] = Pr[(E \wedge F) \vee (\neg E \wedge \neg F)] = 0.875 * .5 + .125 * .5 = .5$
- $Pr_U[D_1(u) = 1] = \frac{1}{2}$
- **QED**

Therefore $Adv^{prg}_{G_1}(D_1) = |Pr_K[D_1(G_1(k)) = 1] - Pr_U[D_1(u) = 1]| = 1 - \frac{1}{2} = \frac{1}{2}$

## 4.2

Bits are 0-indexed, and assume that $l > n = 256$
Let $V_2(s)$ represent the function by which either register in $G_2$ generates a new bit given a register state $s$, where $s$ is a bit string $\in \{0,1\}^n$. $V_2(s) = s_2 \oplus s_{93} \oplus s_{177} \oplus s_{230}$
We can define $D : \{0,1\}^l \to \{0,1\}$ as:

**Algorithm 3:** $D_2(s)$

```
if V_2(s[:n]) == s[n] then
    return 1
end
return 0
```

Now let's calculate $Adv^{prg}_{G_1}(D_1) = |Pr_K[D_1(G_1(k)) = 1] - Pr_U[D_1(u) = 1]|$, where $k$ and $u$ are uniform random variables on $\{0,1\}^{512}$ and $\{0,1\}^l$ respectively

First, a little proof to make our task easier. We claim that $\forall k, k' \in \{0,1\}^n$, the output of $G_2(k||k')$ is equivalent to that of some other LFSR $G'_2(k \oplus k'), G'_2 : \{0,1\}^n \to \{0,1\}^l$ s.t. $G'_2$ uses only one register of $n$ bits, but initializes it using $k \oplus k'$. It uses the same tap function $V_2$:

- $\forall k, k' \in \{0,1\}^n$, $G_2(k||k') = G'_2(k) \oplus G'_2(k')$; this follows from realizing that $G_2$ is two LFSRs running in parallel, with their results XORed
- By our proof from problem 3, we know that $G'_2(k) \oplus G'_2(k') = G'_2(k \oplus k')$
- **QED**

Moving on, we will substitute $G'_2$ for $G_2$, and interpret random variables accordingly
First, we evaluate $Pr_K[D_2(G_2(k)) = 1] = Pr_K[D_2(G'_2(k' = k[:n] \oplus k[n:])) = 1]$

- For any execution of $G'_2$, the $n+1$th bit output by $G'_2$ was generated using the register state initialized by the key $k' = k[:n] \oplus k[n:]$ ($G'_2$ is an LSFR)
- This state is copied into the first $n$ bits generated by $G'_2$. Call this sub string $k''$ ($G'_2$ is an LFSR)
- We observe that $k' = k''$ and therefore $V_2(k') = V_2(k'')$;
- Moreover, since $V_2$ implements the method by which $G'_2$ generates new bits, $\forall s \in \{G'_2(k) : k \in \{0,1\}^n\}, k' = s[:n], V_2(k') = s_n$
- $\forall s \in \{0,1\}^l, k' = s[:n], D_2(s)$ operates by evaluating $V_2(k)' = s_n$ and returning 1 if the equality holds, and 0 otherwise
- Therefore $\forall s \in \{G'_2(k) : k \in \{0,1\}^n\}, D_2(s) = 1$
- This means that $Pr_K[D_2(G'_2(k' = k[:n] \oplus k[n:])) = 1] = 1$
- Therefore $Pr_K[D_2(G_2(k)) = 1] = 1$
- **QED**

Next, we need to evaluate $Pr_U[D_2(u) = 1]$

- We already showed that $D$ returns 1 for all strings produced by $G_2$, but $\{G_2(k) : k \in \{0,1\}^n\} \subseteq U$, so we need to account for false positives returned by $D_2$
- Let $E$ describe the event $u_n = 0$, where $u$ is a uniformly random string on $U$. $Pr_U[E] = Pr_U[\neg E] = \frac{1}{2}$
- Let $F$ describe the event that $V_2(u[:n]) = 0$. Since the bits tapped by $V_2$ are uniformly random, we can count results of a truth table to arrive at a probability for $F$. Doing so, we get $Pr_U[F] = 0.5$ and $Pr_U[\neg F] = 0.5$

- Therefore $Pr_U[D_2(u) = 1] = Pr[(E \wedge F) \vee (\neg E \wedge \neg F)] = 0.5 * .5 + .5 * .5 = .5$
- $Pr_U[D_2(u) = 1] = \frac{1}{2}$

Therefore $Adv^{prg}_{G_2}(D_2) = |Pr_K[D_2(G_2(k)) = 1] - Pr_U[D_2(u) = 1]| = 1 - \frac{1}{2} = \frac{1}{2}$

## 4.3

I know that this question has something to do with doing an analysis on the bytes generated by $G_3$, and that Cash recommended that we use Python to try and look for an initial pattern to base our distinguisher and analysis on. The python script I tried to write in order to do this kept giving me errors I've never experienced in the context of working with lists (I got 'Nonetype object is not subscriptable' errors). I spent an hour just trying to get the script to work, but I failed, so here's my best shot at constructing a pattern from observing one time iterations of $G_3$:
For $D_3(s)$, should return 1 if $s$ contains a series of length 4 made up of two alternating bytes. Otherwise, $D_3$ returns 0
In order to evaluate, $Adv^{prg}_{G_3}(D_3) = |Pr_K[D_3(G_3(k)) = 1] - Pr_U[D_3(u) = 1]|$, I'm going to say that $Pr_K[D_3(G_3(k)) = 1] = 1$, because I didn't observe any other pattern, because my brain was too tired to come up with one and python was being too temperamental with me to be of any help.
I'm going to say that $Pr_U[D_3(u) = 1] = 1/255^2$, since that's the probability of observing two sets of bytes following two sets of equivalent bytes. At least I think that's correct, because this probability grows with the size of the $s$, which is discouraging.
$Adv^{prg}_{G_3}(D_3) = |Pr_K[D_3(G_3(k)) = 1] - Pr_U[D_3(u) = 1]| = 1 - 1/255^2$
I'm sorry I don't have anything more rigorous, my brain just hurts from debugging and this is due soon.