Indian Institute of Technology Delhi

COL 759 - Cryptography

Assignment 1

Securities of Pseudo-Random Generators, Functions and their Relations

Garv Nagori, Naman Agarwal
2021CS10549, 2021CS10555

Contents

§1 Part A

§1.1 Question 1

§1.1.1 Subpart a.

Claim 1.1 — No Symmetric-Key Encryption scheme satisfies Perfect Two Time security

Proof. Perfect Two Time Security Game is defined as follows:

Two Time Security Game

Challenger		**Adversary**
$k \xleftarrow{\$} \{0,1\}^{l_2}$		
$b \xleftarrow{\$} \{0,1\}$		
	$\xleftarrow{(m_{00}, m_{01})}$	
	$\xrightarrow{(m_{10}, m_{11})}$	
	$\xrightarrow{(Enc(k, m_{b0}), Enc(k, m_{b1}))}$	
	$\xleftarrow{b'}$	
	Adversary wins if $b = b'$	

We say that (Enc, Dec) scheme is perfectly Two-Time secure if $\Pr[\mathcal{A} \text{ wins}] = \frac{1}{2}$
To prove no scheme satisfies perfect two-time security we just have to show an Adversary that can win the game with probability $> \frac{1}{2}$
The Adversary is defined as follows:

1. Send two pairs of messages such that $m_{00} = m_{10}$, i.e. send (a, b) and (a, c)
2. Challenger sends back encrypted messages (x, y)
3. Sample k $\xleftarrow{\$} \{0,1\}^{l_2}$
4. Check if $Dec(k, x) = a$
5. If true find $Dec(k, y)$ and return the pair which satisfies the $(Dec(k, x), Dec(k, y))$
6. Else return 1

The winning probability is given by $\frac{1}{2}\Pr[\mathcal{A} \leftarrow 0 \mid \mathcal{C} \leftarrow 0] + \frac{1}{2}\Pr[\mathcal{A} \leftarrow 1 \mid \mathcal{C} \leftarrow 1]$

$$\Pr[\mathcal{A} \leftarrow 0 \mid \mathcal{C} \leftarrow 0] = \frac{1}{2^{l_2}}$$

i.e. the key adversary sampled turned out to be the right key.

$$\begin{aligned}
\Pr[\mathcal{A} \leftarrow 1 \mid \mathcal{C} \leftarrow 1] &= 1 - \Pr[\mathcal{A} \leftarrow 0 \mid \mathcal{C} \leftarrow 1] \\
&= 1 - \frac{1}{2^{l_2}} \cdot \Pr\left[\exists k' \text{ s.t. } \mathrm{Enc}(k', m_{00}) = Enc(k, m_{10}) \text{ and } \mathrm{Enc}(k', m_{01}) = \mathrm{Enc}(k, m_{11})\right] \\
&= 1 - \frac{1}{2^{l_2}} \cdot 1 \cdot \frac{1}{2^{l_1}}
\end{aligned}$$

Hence we finally get the total winning probability of A to be greater than $\frac{1}{2}$. □

§1.1.2 Subpart b.

We define the pairwise independent hash function game, similar to the PRF game, as follows:

Pairwise Independent Hash Function Game

Challenger $\mathcal{C}$		**Adversary $\mathcal{A}$**
$b \leftarrow\$ \{0,1\}$		
$k \leftarrow\$ \{0,1\}^{\mathcal{K}}$		
if $b = 0$ then		
$y_i = h_k(x_i)$	$\xleftarrow{(x_0, x_1)}$	
else $y_0, y_1 \leftarrow\$ \{0,1\}^l$	$\xrightarrow{(y_0, y_1)}$	
	$\xleftarrow{b'}$	
	Adversary wins if $b = b'$	

We define our Encryption scheme as follows:

$$\begin{aligned} \mathrm{Enc}(k, m): &\quad r \leftarrow \{0,1\}^l \\ &\quad \mathrm{ct} = (r, h_k(k \oplus r) \oplus m) \\ \mathrm{Dec}(k, \mathrm{ct}' = (r, \mathrm{ct})): &\quad \mathrm{ct} \oplus h_k(k \oplus r) \end{aligned}$$

Claim 1.2 — Above encryption satisfies $\mathcal{O}(2^{-l})$ perfect security.

World b

Challenger		**Adversary**
$k \leftarrow\$ K$		
	$\xleftarrow{m_b = (m_{b0}, m_{b1})}$	
$ct_0, ct_1 = Enc(k, m_{b0}), Enc(k, m_{b1})$	$\xrightarrow{ct_0, ct_1}$	
	$\xleftarrow{b'}$	
	Adversary wins if $b = b'$	

$$p_b = \Pr[0 \leftarrow \mathcal{A} \mid \text{World b}]$$

Hybrid World b

Challenger		**Adversary**
$k \leftarrow\$ \mathcal{K}$		
	$\xleftarrow{m_b = (m_{b0}, m_{b1})}$	
$r_0, r_1, y_0, y_1 \leftarrow\$ \{0,1\}^l$		
$ct_0, ct_1 = (r_0, y_0 \oplus m_{b0}), (r_1, y_1 \oplus m_{b1})$	$\xrightarrow{ct_0, ct_1}$	
	$\xleftarrow{b'}$	
	Adversary wins if $b = b'$	

$$p_{h,b} = \Pr[0 \leftarrow \mathcal{A} \mid \text{Hybrid World b}]$$

Reduction $\mathcal{B}_b$

Challenger $\mathcal{C}$		**Reduction** $\mathcal{B}_b$		**Adversary** $\mathcal{A}$
$b \leftarrow\$ \{0,1\}$				
$r_0, r_1 \leftarrow\$ \{0,1\}^l$				
	$\xleftarrow{(m_{b0}, m_{b1})}$		$\xleftarrow[(m_{10}, m_{11})]{(m_{00}, m_{01})}$	
if $b = 0$ then				
$k \leftarrow\$ K$ and $y_i = r_i, h_k(r_i)$				
else $y_0, y_1 \leftarrow\$ \{0,1\}^l$	$\xrightarrow{y_0, y_1}$		$\xrightarrow{y_0, y_1}$	
	$\xleftarrow{b'}$		$\xleftarrow{b'}$	
		Adversary wins if $b = b'$		

If $b = 0$, then this is World b, else if $b = 1$ then this is the Hybrid World b.
Here, $\mathsf{PRGAdv}_{\mathcal{A}} = |p_b - p_{hyb,b}|$. If the difference is non-negligible we break the security of $\mathcal{H}$ with non-negligible advantage. So we have created an encryption which satisfies $\mathcal{O}\left(2^{-l}\right)$ perfect security.

§1.2 Question 2

§1.2.1 Subpart a.i.

Claim 1.1 — $\mathcal{G}'$ is not secure against related-key attacks

Breaking Encryption

Challenger		**Adversary**
$b \xleftarrow{\$} \{0,1\}$		
$m_0, m_1 \xleftarrow{\$} \{0,1\}^{2n}$		
if $b = 0$		
$y_0, y_1 = G_n(m_0) \wedge G_n(m_1)$		
else		
$y_0, y_1 \xleftarrow{\$} \{0,1\}^{3n}$		
	$\xrightarrow{\quad y_0, y_1 \quad}$	
		b' = first bit of y_0
	$\xleftarrow{\quad b' \quad}$	
Adversary wins if $b = b'$		

The advantage of the adversary is given by $|\Pr[\mathcal{A} \leftarrow 0 \mid \mathcal{C} \leftarrow 0] - \Pr[\mathcal{A} \leftarrow 0 \mid \mathcal{C} \leftarrow 1]|$
We are given that G_n is a secure PRG, so the output is seemingly random.
Now if we bitwise AND 2 seemingly random strings, the probability that the first bit is 0 will be $\frac{3}{4}$
But if we sample a string, the probability that the first bit is 0 is $\frac{1}{2}$
So we exploit this fact to get an advantage of $\frac{1}{4}$ and hence disprove the security of the encryption.
□

§1.2.2 Subpart a.ii.

Claim 1.2 — $\mathcal{G}'$ is secure against related-key attacks

World 0

Challenger		**Adversary**
$m_0, m_1 \xleftarrow{\$} \{0,1\}^n$		
$y = G_n(m_0) \oplus G_n(m_1)$		
	$\xrightarrow{\quad y \quad}$	
	$\xleftarrow{\quad b' \quad}$	
Adversary wins if $b = b'$		

Hybrid World 1

Challenger	**Adversary**
$m_0 \leftarrow\$ \{0,1\}^{3n}$	
$m_1 \leftarrow\$ \{0,1\}^{n}$	
$y = m_0 \oplus G_n(m_1)$	
$\xrightarrow{\quad y \quad}$	
$\xleftarrow{\quad b' \quad}$	
Adversary wins if $b = b'$	

Hybrid World 2

Challenger	**Adversary**
$m_0, m_1 \leftarrow\$ \{0,1\}^{3n}$	
$y = m_0 \oplus m_1$	
$\xrightarrow{\quad y \quad}$	
$\xleftarrow{\quad b' \quad}$	
Adversary wins if $b = b'$	

World 1

Challenger	**Adversary**
$y \leftarrow\$ \{0,1\}^{3n}$	
$\xrightarrow{\quad y \quad}$	
$\xleftarrow{\quad b' \quad}$	
Adversary wins if $b = b'$	

We have created two hybrid worlds and claim that consecutive worlds are indistinguishable from each other.

- World 0 and Hybrid World 0 are indistinguishable based on the fact that G is a secure pseudorandom generator (PRG).
- Similarly, Hybrid World 0 and Hybrid World 1 are indistinguishable.
- Hybrid World 1 and World 1 are indistinguishable due to the property that XORing two random strings will generate a seemingly random string.
  This can be understood by the fact that the probability that a bit is 0 or 1 will be $\frac{1}{2}$ each after XORing.

□

§1.2.3 Subpart b.i.

Claim 1.3 — $\mathcal{F}'$ is not secure against related-key attacks

Breaking Encryption

Challenger		**Adversary**
$b \leftarrow\$ \{0,1\}$		
$k \leftarrow\$ \{0,1\}^n$	$\xleftarrow{(m,m)}$	
if $b = 0$ $y = F_n(k,m) \oplus F_n(k,m)$ else $y \leftarrow\$ \{0,1\}^n$		
	$\xrightarrow{y}$	
		$if y = 0 : b' = 0$ $else : b' = 1$
	$\xleftarrow{b'}$	
	Adversary wins if $b = b'$	

The advantage of the adversary is given by $|\Pr[\mathcal{A} \leftarrow 0 \mid \mathcal{C} \leftarrow 0] - \Pr[\mathcal{A} \leftarrow 0 \mid \mathcal{C} \leftarrow 1]|$
$\frac{1}{2}\Pr[\mathcal{A} \leftarrow 0 \mid \mathcal{C} \leftarrow 0] = 1$ as $F_n(k,m) \oplus F_n(k,m)$ will always be 0.
$\frac{1}{2}\Pr[\mathcal{A} \leftarrow 0 \mid \mathcal{C} \leftarrow 1] = \frac{1}{2^n}$
So a non-negligible advantage adversary exists and hence the encryption is not secure.

□

§1.2.4 Subpart b.ii.

Claim 1.4 — $\mathcal{F}'$ is secure against related-key attacks

World 0

Challenger		**Adversary**
$b \leftarrow\$ \{0,1\}$		
$k \leftarrow\$ \{0,1\}^n$		
	$\xleftarrow{m}$	
$ct = F_n k, m \oplus m$		
	$\xrightarrow{ct}$	

Hybrid World

Challenger		**Adversary**
$b \leftarrow\$ \{0,1\}$		
	$\xleftarrow{m}$	
$y \leftarrow\$ \{0,1\}^n$ $ct = y \oplus m$		
	$\xrightarrow{ct}$	

World 1

Challenger		**Adversary**
$b \leftarrow\$ \{0,1\}$		
	$\xleftarrow{\quad m \quad}$	
$y \leftarrow\$ \{0,1\}^n$		
	$\xrightarrow{\quad y \quad}$	

World 0 and Hybrid World are indistinguishable based on the fact that F_n is a secure PRF
Hybrid World and World 1 are indistinguishable by the fact that XORing a random string to our message will be seemingly random.
This can be intuitively understood by Shannon's OTP where we XORed our key to the message and the cipher-text appeared to be random with respect to our message.
Hence, by the hybrid world argument, this is a secure PRF

□

§1.3 Question 3

§1.3.1 Subpart a.

Given $\mathcal{G} = \{G_n : \{0,1\}^n \to \{0,1\}^{3n}\}$, we construct $\mathcal{G}' = \{G'_n : \{0,1\}^n \to \{0,1\}^{3n}\}$ such that Let $x = x_0||x_1$ where $|x_0| = \left\lfloor \frac{|x|}{2} \right\rfloor$ and $|x_1| = \left\lceil \frac{|x|}{2} \right\rceil$ then,

$$G'_n(x) = G_{|x_0|}(x_0) \,||\, G_{|x_1|}(x_1)$$

§1.3.2 Subpart b.

Claim 1.1 — $\mathcal{G}'$ is not secure against related-key attacks

Proof. The Related-Key Attack Game is given as:

Related Key Attack Game

Challenger		**Adversary**
$s \leftarrow\$ \{0,1\}^n$		
$b \leftarrow\$ \{0,1\}$		
$if\ (b = 0)\ y = G_n(s)$		
$else\ y \leftarrow\$ \{0,1\}^{3n}$		
$s \leftarrow (s+1) \mod 2^n$	$\xrightarrow{\quad y \quad}$	Polynomially many Queries
	$\xleftarrow{\quad b' \quad}$	
	Adversary wins if $b = b'$	

The Adversary is defined as follows:

1. Query 3 times and get (a_0, a_1), (b_0, b_1) and (c_0, c_1)
2. Send 0 if $a_0 = b_0$ or $b_0 = b_1$
3. Send 1 otherwise

The adversary's winning probability is given by

$$\Pr[\mathcal{A}\text{wins}] = \frac{1}{2} \cdot \Pr[\mathcal{A} \leftarrow 0 \,|\, \mathcal{C} \leftarrow 0] + \frac{1}{2} \cdot \Pr[\mathcal{A} \leftarrow 1 \,|\, \mathcal{C} \leftarrow 1]$$

$$\Pr[\mathcal{A} \leftarrow 0 \,|\, \mathcal{C} \leftarrow 0] = 1$$

This comes from the fact that if $b = 0$ then either $a_0 = b_0$ because adding 1 to s doesn't change the first $\lfloor \frac{n}{2} \rfloor$ bits in most cases. It only changes if the last $\lceil \frac{n}{2} \rceil$ bits were 1, in which case, we compare $b_0 = c_0$ to be sure.

$$\begin{aligned}
\Pr[\mathcal{A} \leftarrow 1 \,|\, \mathcal{C} \leftarrow 1] &= 1 - \Pr[\mathcal{A} \leftarrow 0 \,|\, \mathcal{C} \leftarrow 1] \\
&= 1 - \Pr[a_0 = b_0 \text{ or } b_0 = c_0 \text{ given they are random bitstrings }] \\
&= 1 - 2 \cdot \frac{2^{\lfloor \frac{n}{2} \rfloor}}{2^n} \\
&= 1 - \frac{2}{2^{\lceil \frac{n}{2} \rceil}}
\end{aligned}$$

Hence, the winning probability of Adversary is $\Pr[\mathcal{A} \text{ wins}] = 1 - \frac{2}{2^{\lceil \frac{n}{2} \rceil}}$, and $\Pr[\mathcal{A} \text{ wins}] - \frac{1}{2} = \frac{1}{2} - \frac{2}{2^{\lceil \frac{n}{2} \rceil}}$ is non-negligible □

§1.3.3 Subpart c.

Claim 1.2 — If there exists a PPT adversary $\mathcal{A}$ that breaks PRGsecurity of $\mathcal{G}'$, then there exists a PPTreduction $\mathcal{B}$ that breaks the PRG Security of $\mathcal{G}$

Proof. The PRG security game is given by

PRG Security Game

Challenger	**Adversary**
$b \leftarrow\$ \{0,1\}$	
if $(b = 0)$ then	
$s \leftarrow\$ \{0,1\}^n$ and $y = G'(s)$	
else $y \leftarrow\$ \{0,1\}^{3n}$	
	$\xrightarrow{\quad y \quad}$
	$\xleftarrow{\quad b' \quad}$
Adversary wins if $b = b'$	

We define this game as two worlds.
We then can create intermediate hybrid worlds that break the PRG Security of G

World 0

Challenger	**Adversary**
$b = 0$	
$s \leftarrow\$ \{0,1\}^n$ and $y = G'(s)$	
	$\xrightarrow{\quad y \quad}$
	$\xleftarrow{\quad b' \quad}$
Adversary wins if $b = b'$	

$p_0 = \Pr[0 \leftarrow \mathcal{A} \,|\, \text{World } 0]$

World 1

Challenger	**Adversary**
$b = 1$	
$y \leftarrow\$ \{0,1\}^{3n}$	
	$\xrightarrow{\quad y \quad}$
	$\xleftarrow{\quad b' \quad}$
Adversary wins if $b = b'$	

$p_1 = \Pr[0 \leftarrow \mathcal{A} \,|\, \text{World } 1]$
We define a hybrid world as follows:

Hybrid World

Challenger		**Adversary**
$s_0 \leftarrow\$ \{0,1\}^{\lfloor \frac{3n}{2} \rfloor}$		
$s_1' \leftarrow\$ \{0,1\}^{\lceil \frac{n}{2} \rceil}$		
$y = s_0 \,\|\| \, G(s_1')$		
	$\xrightarrow{\quad y \quad}$	
	$\xleftarrow{\quad b' \quad}$	
Adversary wins if $b = b'$		

$p_{hyb} = \Pr[0 \leftarrow \mathcal{A} \,|\, \text{Hybrid World}]$

Claim 1.3 — If there exists a PPT $\mathcal{A}$ that distinguishes between World 0 and Hybrid World with non-negligible probability, there exists a PPT $\mathcal{B}$ that breaks PRG security of $\mathcal{G}$

Let us call this reduction $\mathcal{B}_0$. The reduction works as follows:

Reduction $\mathcal{B}_0$

Challenger $\mathcal{C}$		**Reduction** $\mathcal{B}_0$		**Adversary** $\mathcal{A}$
$b \leftarrow\$ \{0,1\}$				
if $b = 0$ then		$s_1' \leftarrow\$ \{0,1\}^n$		
$s \leftarrow\$ \{0,1\}^n$ and $y = G(s)$				
else $y \leftarrow\$ \{0,1\}^{3n}$	$\xrightarrow{\quad y \quad}$		$\xrightarrow{\quad y \| \|G(s_1') \quad}$	
	$\xleftarrow{\quad b' \quad}$		$\xleftarrow{\quad b' \quad}$	
		Adversary wins if $b = b'$		

If $b = 0$, then this is World 0, else if $b = 1$ then this is the Hybrid World.
Here, $\mathsf{PRGAdv}_{\mathcal{A}} = |p_0 - p_{hyb}|$. If the difference is non-negligible we break the PRG security of $\mathcal{G}$ with non-negligible advantage.

Claim 1.4 — If there exists a PPT $\mathcal{A}$ that distinguishes between World 1 and Hybrid World with non-negligible probability, there exists a PPT $\mathcal{B}$ that breaks PRG security of $\mathcal{G}$

Let us call this reduction $\mathcal{B}_1$. The reduction works similar as above, as follows:

Reduction $\mathcal{B}_1$

Challenger $\mathcal{C}$		**Reduction** $\mathcal{B}_1$		**Adversary** $\mathcal{A}$
$b \leftarrow\$ \{0,1\}$				
if $b = 0$ then		$s_1 \leftarrow\$ \{0,1\}^{3n}$		
$s \leftarrow\$ \{0,1\}^n$ and $y = G(s)$				
else $y \leftarrow\$ \{0,1\}^{3n}$	$\xrightarrow{\quad y \quad}$		$\xrightarrow{\quad s_1 \| y \quad}$	
	$\xleftarrow{\quad b' \quad}$		$\xleftarrow{\quad b' \quad}$	
		Adversary wins if $b = b'$		

If $b = 0$, then this is the Hybrid World, else if $b = 1$ then this is World 1.
Here, $\mathsf{PRGAdv}_{\mathcal{A}} = |p_{hyb} - p_1|$. If the difference is non-negligible we break the PRG security of $\mathcal{G}$ with non-negligible advantage.
Coming back to our original claim, we construct reduction $\mathcal{B}$ such that it randomly chooses any one $\mathcal{B}_0$ and $\mathcal{B}_1$ with probability 0.5
In this case,
$\Pr[0 \leftarrow \mathcal{A} \mid 0 \leftarrow \mathcal{C}] = \frac{1}{2} \cdot (p_0 + p_{hyb})$ and $\Pr[0 \leftarrow \mathcal{A} \mid 1 \leftarrow \mathcal{C}] = \frac{1}{2} \cdot (p_{hyb} + p_1)$
Hence, the advantage is $\frac{1}{2} \cdot |p_0 - p_1|$ which is non-negligible if $|p_0 - p_1|$ is non-negligible. □

§1.4 Question 4

§1.4.1 Subpart a.

We can model the PRF game (with only test queries) using two worlds, one where the challenger picks 0 or a pseudo-random function and the other where the challenger picks 1 or a truly random function.

World 0

Challenger		**Adversary**
$b = 0$		
$k \leftarrow\$ \{0,1\}^n$		
	$\xleftarrow{\quad x \quad}$	Polynomially
$y = F_n(k, x)$	$\xrightarrow{\quad y \quad}$	many queries
	$\xleftarrow{\quad b' \quad}$	
	Adversary wins if $b = b'$	

$$p_0 = \Pr[0 \leftarrow \mathcal{A} \mid \text{World } 0]$$

Taking a truly random function is the same as sampling a new value of $y = F(x)$ whenever a new x is queried.

World 1

Challenger		**Adversary**
$b = 1$		
	$\xleftarrow{\quad x \quad}$	Polynomially
$y \leftarrow\$ \{0,1\}^n$	$\xrightarrow{\quad y \quad}$	many queries
	$\xleftarrow{\quad b' \quad}$	
	Adversary wins if $b = b'$	

$$p_1 = \Pr[0 \leftarrow \mathcal{A} \mid \text{World } 1]$$

To prove that this is a secure PRG we create intermediate hybrid worlds.
We define the i^{th} Hybrid World as follows:

Hybrid World i

Challenger		**Adversary**
$s_0, s_1 \leftarrow\$ \{0,1\}^n$		
	$\xleftarrow{\quad x \quad}$	Polynomially
$y = F_n(s_{x[i]}, \text{last } n-i \text{ bits of } x)$	$\xrightarrow{\quad y \quad}$	many queries
	$\xleftarrow{\quad b' \quad}$	
	Adversary wins if $b = b'$	

$$p_{hyb}^{(i)} = \Pr[0 \leftarrow \mathcal{A} \mid \text{Hybrid World } i]$$

In this hybrid world, we only use the last i bits of x to create y.
For example, the difference between World 0 and Hybrid World 1 is at the first step. In world 0 we calculate $G(k) = s_0, s_1$ and then use the first bit of x to determine which part to use ahead. In world 1 we choose s_0, s_1 randomly instead of calculating $G(k)$. That means the only difference is the selection of s_0, s_1 is done pseudo-randomly or truly at random.
Between any two consecutive hybrid worlds, the only difference is that in one we have used a PRG , and the other we chose it truly at random.
At each hybrid world, we move one layer down and make it truly random. Therefore, there will be $log(n) - 1$ hybrid worlds and $log(n)$ reductions.
Let the advantage in PRG game be given by $\mathsf{PRGAdv}_{\mathcal{A}}$. This advantage would be negligible assuming the security of $\mathcal{G}$. So any reduction $\mathcal{B}_i$ between Hybrid world i and Hybrid World $i+1$, will have an advantage of $\mathsf{PRGAdv}_{\mathcal{A}} = p_{hyb}^{i} - p_{hyb} i + 1$. The overall advantage would then be,

$$\begin{aligned} |p_0 - p_1| &= p_0 - p_{hyb}^1 + p_{hyb}^1 - p_{hyb}^2 + \cdots + p_{hyb}^i - p_{hyb}^{i+1} + \cdots - p_1 \\ &= log(n) \cdot \mathsf{PRGAdv}_{\mathcal{A}} \end{aligned}$$

This advantage is negligible because $\mathsf{PRGAdv}_{\mathcal{A}}$ is negligible and multiplying by a factor of $log(n)$ doesn't make it non-negligible.
We prove a more generalized version formally in the next part.
Hence, if $\mathcal{G}$ is a secure PRG , $\mathcal{F}$ is a secure PRF.

§1.4.2 Subpart b.

The proof from the above part will also work here. Here, instead of $log(n) - 1$ hybrid worlds, there will be $n - 1$ hybrid worlds, and consequently n reductions.
Following similarly as above we get the advantage of this as $n \cdot \mathsf{PRGAdv}_{\mathcal{A}}$.

> **Claim 1.1 —** Let $f(n)$ be a negligible function. For any polynomial function $p(n)$, $p(n) \cdot f(n)$ is also negligible.

Proof. Recalling the definition of a negligible function,
A negligible function is a function $f : \mathbb{N} \to \mathbb{R}$ such that $\forall c \exists n_0 such that \forall n > n_0$,

$$|f(n)| < \frac{1}{n^c}$$

Now, for $p(n)$, we can find a d such that $\exists n_0 such that \forall n > n_0, p(n) < n^d$.
So, $p(n) \cdot f(n) < n^d \cdot f(n)$.
If it is negligible, we have $\forall c, n^d \cdot f(n) < \frac{1}{n^c}$ or in other words,

$$\forall c, f(n) < \frac{1}{n^{c+d}}.$$

We know this to be true since $f(n)$ is a negligible function so there will always exist a c. Hence, $p(n) \cdot f(n)$ is negligible.

□

This shows that the advantage for $\{0,1\}^n$ (also, $\{0,1\}^{log(n)}$ is still negligible, and so $\mathcal{F}$ is a secure PRF family.

§1.4.3 Subpart c

This scheme is not secure because of an extension attack.
The Adversary $\mathcal{A}$ works as follows:

1. Send x to the challenger and receive y_0

2. Send $x \,||\, 1$ to the challenger and receive y_1

3. Let $s_0, s_1 = G(y_0)$. If $s_1 = y_1$ return 1, else return 0.

This attack works because $\mathcal{F}$ is deterministic so when it calculates $G(x \,||\, 1)$, it first calculates $\mathcal{F}(x)$ and then applies G. We can easily check for this case.
The Adversary's winning probability can be calculated as $\frac{1}{2}\Pr[\mathcal{A} \leftarrow 0 \mid \mathcal{C} \leftarrow 0] + \frac{1}{2}\Pr[\mathcal{A} \leftarrow 1 \mid \mathcal{C} \leftarrow 1]$

$$\Pr[\mathcal{A} \leftarrow 0 \mid \mathcal{C} \leftarrow 0] = 1$$

This is because whenever we use the pseudo-random function, it always calculates $F(x \,||\, 1)$ based on $F(x)$ and we exploit it to our own advantage.

$$\begin{aligned}\Pr[\mathcal{A} \leftarrow 1 \mid \mathcal{C} \leftarrow 1] &= 1 - \Pr[\mathcal{A} \leftarrow 0 \mid \mathcal{C} \leftarrow 1] \\ &= 1 - \Pr[\text{Randomly sampled strings satisfies the property}] \\ &= 1 - \frac{2^n}{2^{2n}} = 1 - \frac{1}{2^n}\end{aligned}$$

The adversary's winning probability is thus given by, $\Pr[[]\mathcal{A} \text{ wins}] = 1 - \frac{1}{2^{n+1}}$.
This is non-negligible for the PRF game.

§1.4.4 Subpart d.

To make it possibly secure, we use $\mathcal{F}$ to create $\mathcal{F}'$.

$$F'(k, x) = F(k, G_{|x|}(x))$$

Here, the scheme is safe against the attack given in the previous part because there is no direct correlation between $G(x)$ and $G(x \,||1)$.
To get related pairs, we want another input x' such that $G(x') = G(x) \,||\, 1$.
Finding such a x' is a very difficult task becuase PRGs are One-Way Functions (OWFs), meaning finding an input for a specific output (or "taking inverse") is very difficult.
Hence, this scheme is plausibly secure.

§2 Part B

§2.1 Coding Question 1.

We use the fact that if a sub-string of the cookie is present in the message then the length of encryption is smaller than other strings.
So we check character by character to generate the cookie. If there is only one character that gives the minimum length string then we append that character to our cookie string.
We found some edge cases when multiple characters appended to our message give the same encryption length irrespective of whether or not it is a substring. One such example was when cookie used is "aaaaaaaaaaaaaaaaaaaaaaa" and when our message was "aaaaaaaaaaaaaaa" appending any character to this gave the same encrption length. This might be because of the encryption scheme used and so to handle this we added a while loop in our code. In that we removed the first character of the message and then followed the same procedure as above to identify the next correct character.

§2.2 Coding Question 2.

Breaking 2DES with two keys (k_1, k_2), message m and ciphertext ct. The adversary is given m and ct and has to find out (k_1, k_2).
Normally this would require $\mathcal{O}\left(|\mathcal{K}|^2\right)$ time to iterate over all possible key pairs, but we can use "Meet in the middle" technique to create an attack that runs in $\mathcal{O}(|\mathcal{K}|)$ time but requires $\mathcal{O}(|\mathcal{K}|)$ space to work.
We iterate over all keys and store $\mathsf{Enc}(k, m)$ and the corresponding key k in a database (dictionary in Python) such that finding a possible match is faster than a linear search. The size of the database is $\mathcal{O}(|\mathcal{K}|)$
Next, we again iterate over all possible keys, but this time we calculate $\mathsf{Dec}(k, ct)$ and search for it in the database.
If there is a match we return the k_1 as the key that generated the value in the database, and k_2 as the key that decrypted ct to give the value in the database.
This attack shows that encrypting twice using different keys is not as good as one might think since the adversary comes up with better techniques in response.

§3 Acknowledgements

We have used the style file from here[1] to typeset and the style file from jhere[2] for cryptographic games and protocols to produce this document.

[1] https://github.com/vEnhance/dotfiles/blob/main/texmf/tex/latex/evan/evan.sty
[2] https://github.com/arnomi/cryptocode