TD 3: Security Assumptions (corrected version)

Exercise 1. *Advantage(s)*

We consider two distributions D_0 and D_1 over $\{0,1\}^n$ and the following experiment.

$\mathcal{C}$	$\mathcal{A}$
sample $b \hookleftarrow U(0,1)$ sample $x \hookleftarrow D_b$ send x to $\mathcal{A}$	
	compute a bit b' send b' to $\mathcal{C}$
If $b = b'$, say "Win", else say "Lose".	

We say that a PPT (Probabilistic, Polynomial-Time) algorithm $\mathcal{A}$ is a *distinguisher* if there exists a non-negligible ε such that, in this experiment, $\Pr[\texttt{Win}] \geq 1/2 + \varepsilon$. The distributions D_0 and D_1 are said to be *indistinguishable* if there is no such distinguisher.

1. Show that this definition of indistinguishability is equivalent to the one seen during the lecture.

   ☞ We will answer by giving two reductions.

   Let $\mathcal{A}$ be an attacker regarding the TD definition; we then consider $\mathcal{A}'$, defined exactly as $\mathcal{A}$ except that it outputs b' rather than sending it to $\mathcal{C}$. We have

$$\begin{aligned}
\mathrm{Adv}(\mathcal{A}') &= |\Pr[\mathcal{A}' \xrightarrow{Exp_0} 1] - \Pr[\mathcal{A}' \xrightarrow{Exp_1} 1]| \\
&= |\Pr[b' = 1 | b = 0] - \Pr[b' = 1 | b = 1]| \\
&= |1 - \Pr[b' = 0 | b = 0] - \Pr[b' = 1 | b = 1]| \\
&= |1 - 2\Pr[b' = 0 \text{ and } b = 0] - 2\Pr[b' = 1 \text{ and } b = 1]| \\
&= |1 - 2\Pr(\texttt{Win})| \geq 2\varepsilon.
\end{aligned}$$

   Conversely, let $\mathcal{A}'$ be an attacker regarding the definition seen in class, and define $\mathcal{A}$ exactly as $\mathcal{A}'$ except that, instead of outputting b', it sends it to C. Then, we have

$$\begin{aligned}
\Pr(\texttt{Win}) &= \Pr[b' = 0 \text{ and } b = 0] + \Pr[b' = 1 \text{ and } b = 1] \\
&= \frac{1}{2}(\Pr[b' = 0 | b = 0] + \Pr[b' = 1 | b = 1]) \\
&= \frac{1}{2}(1 - \Pr[b' = 1 | b = 0] + \Pr[b' = 1 | b = 1]) \\
&= \frac{1}{2}(1 + \mathrm{Adv}(\mathcal{A}')).
\end{aligned}$$

   where we have assumed that $\Pr[b' = 1 | b = 1] \geq \Pr[b' = 1 | b = 0]$. Otherwise, we define $\mathcal{A}$ exactly as $\mathcal{A}'$ except that instead of outputting b' it sends $1 - b'$ to C (so that $\Pr[b' = 1 | b = 1] - \Pr[b' = 1 | b = 0]$ is always positive and equal to $\mathrm{Adv}(\mathcal{A}')$).

   Remark. For the reduction to work, we do not need to determine whether $\Pr[\mathcal{A}' \xrightarrow{Exp_0} 1] - \Pr[\mathcal{A}' \xrightarrow{Exp_1} 1]$ is positive or negative. We just need to prove that whatever the case is, there exists an attacker $\mathcal{A}$.

2. A rebellious student decides to define a distinguisher as a PPT algorithm $\mathcal{A}$ with $\Pr[\texttt{Win}] \leq 1/2 - \varepsilon$ in the above experiment (rather than $\geq 1/2 + \varepsilon$). Is this a revolutionary idea?

   ☞ This is again equivalent to the previous definitions. Given an attacker $\mathcal{A}$ for the game defined in this tutorial, we define $\mathcal{A}'$ which acts exactly as $\mathcal{A}$ except that it sends $1 - b'$ to C instead of b'. Then we have $\Pr[\mathcal{A} \texttt{ win}] = 1 - \Pr[\mathcal{A}' \texttt{ win}]$. And so $\Pr[\mathcal{A} \texttt{ win}] \geq 1/2 + \varepsilon$ if and only if $\Pr[\mathcal{A}' \texttt{ win}] \leq 1/2 - \varepsilon$.

Exercise 2. *Around the DDH assumption*

We recall the definition of the DDH assumption.

Definition 1 (Decisional Diffie-Hellman distribution). *Let $\mathbb{G}$ be a cyclic group of (prime) order p, and let g be a public generator of $\mathbb{G}$. The* decisional Diffie-Hellman *distribution (DDH) is, $D_{\text{DDH}} = (g^a, g^b, g^{ab}) \in \mathbb{G}^3$ with a, b sampled independently and uniformly in $\mathbb{Z}/p\mathbb{Z} =: \mathbb{Z}_p$.*

Definition 2 (Decisional Diffie-Hellman assumption). *The decisional Diffie-Hellman assumption states that there exists no probabilistic polynomial-time distinguisher between D_{DDH} and (g^a, g^b, g^c) with a, b, c sampled independently and uniformly at random in $\mathbb{Z}_p$.*

1. Does the DDH assumption hold in $\mathbb{G} = (\mathbb{Z}_p, +)$ for $p = \mathcal{O}(2^\lambda)$ prime? ☞ No. In this case, the D_{DDH} distribution is $(a \cdot g, b \cdot g, (ab) \cdot g)$. This can be distinguished from (ag, bg, cg) by computing the inverse of g (find a Bézout identity $gu + pv = 1$ in logarithmic time), retrieving a, b and c and checking whether $ab = c$ or not. This is always the case in the DDH, and the case with probability $1/p$ in the uniform case. The advantage of a distinguisher returning the boolean value of $ab = c$ is then $\frac{p-1}{p}$.

2. Same question for $\mathbb{G} = (\mathbb{Z}_p^\star, \times)$ of order $p - 1$, with p an odd prime.

   ☞ No, because $p - 1$ (the order the group) is divisible by 2.

   We know that $x^{\frac{p-1}{2}} = 1$ if $x \in \mathbb{Z}_p$ is a square and -1 otherwise (it is actually the Legendre symbol: $\left(\frac{x}{p}\right)$ and can be efficiently computed). So $\left(\frac{g^a}{p}\right)$ gives us the parity of a, that is $\left(\frac{g^a}{p}\right) = 1$ if a is even and $\left(\frac{g^a}{p}\right) = -1$ if a is odd. Hence, if a is uniformly sampled in $\{0, \cdots, p-1\}$ (meaning that g^a is uniformly sampled in $\mathbb{G}$), then $\left(\frac{g^a}{p}\right)$ is uniformly distributed in $\{-1, 1\}$. But in the case of the DDH distribution, if a or b is even, then ab must be even too (or equivalently, it g^a or g^b is a square, then g^{ab} should be a square too). In the same way, if both a and b are odd, then ab must be odd.

   This enables us to build the following distinguisher $\mathcal{A}$:

   - Return DDH if $\left(\frac{g^{ab}}{p}\right)$ is consistent with $\left(\frac{g^a}{p}\right)$ and $\left(\frac{g^b}{p}\right)$ (i.e. ab is odd and both a and b are odd, or ab is even and a or b is even);
   - Return Unif otherwise.

   Let us now compute the advantage of such a distinguisher.

$$\begin{aligned}\text{Adv}^{DDH}(\mathcal{A}) &= |\Pr[\mathcal{A} \to DDH \mid DDH] - \Pr[\mathcal{A} \to DDH \mid Unif]| \\ &= |1 - \Pr[\mathcal{A} \to DDH \mid Unif]|\end{aligned}$$

   Our distinguisher returns Unif only is c is odd and either a or b is even of if c is even and both a and b are odd. But we have seen that these cases could not appear in the DDH distribution. So we have that $\Pr[\mathcal{A} \to DDH \mid DDH] = 1$.

   It then remains to compute $\Pr[A \to DDH \mid Unif]$. Given a Unif instance (g^a, g^b, g^c), we have seen that $\left(\frac{g^a}{p}\right)$, $\left(\frac{g^b}{p}\right)$ and $\left(\frac{g^c}{p}\right)$ are uniform in $\{-1, 1\}$ because a, b, c are uniform in $\{0, \cdots, q-1\}$. They are also independent because a, b and c are. So all eight possibilities for $\left(\left(\frac{g^a}{p}\right), \left(\frac{g^b}{p}\right), \left(\frac{g^c}{p}\right)\right)$ have the same probability and we have

$$\begin{aligned}\Pr[\mathcal{A} \to DDH \mid Unif] &= \Pr[\left(\left(\frac{g^a}{p}\right), \left(\frac{g^b}{p}\right), \left(\frac{g^c}{p}\right)\right) = (1,1,1) \text{ or } (1,-1,1) \text{ or } (-1,1,1) \text{ or } (-1,-1,-1)] \\ &= \tfrac{4}{8} = \tfrac{1}{2}\end{aligned}$$

   To conclude, we have $\text{Adv}(\mathcal{A}) = \frac{1}{2}$, which is non-negligible.

   It remains to show that our distinguisher is PPT. This is the case because it just needs to compute $\left(\frac{h}{p}\right) = h^{\frac{p-1}{2}}$ for three different elements h of $\mathbb{G}$. Computing $h^{\frac{p-1}{2}}$ can be done by fast exponentiation, resulting in at most $\log(p)$ multiplications in $\mathbb{Z}_p$. Each such multiplication takes a time polynomial in $\log(p)$, and so our distinguisher $\mathcal{A}$ is indeed polynomial time (in $\log(p)$).

   Remark. The same reasoning can be adapted if the cardinality of the cyclic group $\mathbb{G}$ is $n = km$ for some small k (and any m). In that case, we would have that $(g^a)^m$ is uniformly distributed among $\{1, g^m, g^{2m}, \cdots, g^{(k-1)m}\}$ if a is uniform, and computing $(g^a)^m$ gives us the value of $a \mod k$. We then can check whether $a \mod k$ and $b \mod k$ are coherent with $ab \mod k$. In the DDH case, this will always be coherent whereas in the uniform case, this will be coherent only with probability $\frac{k-1}{k}$. We hence obtain a distinguisher with advantage $\frac{k-1}{k}$ (which is non negligible for any $k \geq 2$) and whose computation time is $\Theta(k \cdot \text{poly}(\log(n)))$. So if k is polynomial in $\log(n)$, this gives us a polynomial time distinguisher with non negligible advantage. This is why, in the next question, we consider of group of prime cardinality.

   But this implies knowing a factorisation of n.

Exercise 3. *Attacking the DLG problem*

Let $\mathbb{G}$ be a cyclic group generated by g, of (known) prime order p, and let h be an element of $\mathbb{G}$. Let $F : \mathbb{G} \to \mathbb{Z}_p$ be a nonzero function, and let us define the function $H : \mathbb{G} \to \mathbb{G}$ by $H(\alpha) = \alpha \cdot h \cdot g^{F(\alpha)}$. We consider the following algorithm (called *Pollard ρ Algorithm*).

Pollard ρ Algorithm

Input: $h, g \in \mathbb{G}$

Output: $x \in \{0, \ldots, p-1\}$ such that $h = g^x$ or FAIL.

1. $i \leftarrow 1$
2. $x \leftarrow 0, \alpha \leftarrow h$
3. $y \leftarrow F(\alpha); \beta \leftarrow H(\alpha)$
4. **while** $\alpha \neq \beta$ do
5. $\quad x \leftarrow x + F(\alpha) \bmod p; \alpha \leftarrow H(\alpha)$
6. $\quad y \leftarrow y + F(\beta) \bmod p; \beta \leftarrow H(\beta)$
7. $\quad y \leftarrow y + F(\beta) \bmod p; \beta \leftarrow H(\beta)$
8. $\quad i \leftarrow i + 1$
9. **end while**
10. **if** $i < p$ **then**
11. $\quad$ **return** $(x - y)/i \bmod p$
12. **else**
13. $\quad$ **return** FAIL
14. **end if**

To study this algorithm, we define the sequence (γ_i) by $\gamma_1 = h$ and $\gamma_{i+1} = H(\gamma_i)$ for $i \geqslant 1$.

1. Show that in the **while** loop from Steps 4 to 9 of the algorithm, we have $\alpha = \gamma_i = g^x h^i$ and $\beta = \gamma_{2i} = g^y h^{2i}$.

☞ We check these identities by induction on $i \geq 1$. For $i = 1$, they are satisfied since from lines 1 to 3 of the algorithm, we have $x = 0, \alpha = h, y = F(h)$, and $\beta = H(h) = g^y h^2$.

Now, let $i \geqslant 1$ and denote by $x_i, \alpha_i, y_i, \beta_i$ the values taken by x, α, y, β at the beginning of the i-th iteration of the **while** loop. We assume that the identities $\alpha_i = \gamma_i = g^{x_i} h^i$ and $\beta_i = \gamma_{2i} = g^{y_i} h^{2i}$ hold.

At the end of the i-th iteration (or the beginning of the $i+1$-th), we have $x_{i+1} = x_i + F(\alpha_i) \bmod p$, and $\alpha_{i+1} = H(\alpha_i) = \alpha_i \cdot h \cdot g^{F(\alpha_i)} = (g^{x_i} \cdot h^i) \cdot h \cdot g^{F(\alpha_i)} = g^{x_i + F(\alpha_i)} \cdot h^{i+1} = g^{x_{i+1}} \cdot h^{i+1}$. We also have $\beta_{i+1} = H(H(\beta_i)) = H(\beta_i \cdot h \cdot g^{F(\beta_i)}) = \beta_i \cdot h^2 \cdot g^{F(\beta_i)} \cdot g^{F(H(\beta_i))} = g^{y_i} \cdot g^{F(\beta_i)} \cdot g^{F(H(\beta_i))} \cdot h^{2i+2}$, and $y_{i+1} = y_i + F(\beta_i) + F(H(\beta_i))$, hence the identity $\beta_{i+1} = g^{y_{i+1}} h^{2i+2}$.

2. Show that if this loop terminates with $i < p$, then the algorithm returns the discrete logarithm of h in basis g.

☞ When the loop finishes, we have $\alpha = \beta$ and according to Question 1, this gives $g^x h^i = g^y h^{2i}$, thus $h^i = g^{x-y}$. If furthermore the loop finishes with $i < p$ (note that $i > 0$), then since p is prime, i is invertible modulo p and $h = g^u$ where $u = (x - y)/i \bmod p$.

3. Let j be the smallest integer such that there exists $k < j$ such that $\gamma_j = \gamma_k$. Show that $j \leqslant p + 1$ and that the loop ends with $i < j$.

☞ The sequence (γ_i) has its values in the finite group $\mathbb{G}$ of cardinality p. By the pigeonhole principle, there exist two indices $k < j \leqslant p + 1$ such that $\gamma_k = \gamma_j$; then, since (γ_i) is defined by $\gamma_{i+1} = H(\gamma_i)$, this sequence repeats with period a divisor of $j - k$.

Remark: we have $\gamma_{k+t} = \gamma_{j+t}$ for any integer $t \geqslant 0$. This leads to representing the values of the sequence in a shape which looks like the letter ρ, hence the name of the algorithm.

From Question 1, we see that the algorithm simultaneously computes the values of γ_i and γ_{2i} and returns the first index i for which $\gamma_i = \gamma_{2i}$. Since the sequence repeats with period $j - k$, considering the smallest multiple i of $j - k$ that is greater or equal to k, namely $i = (j - k)\lceil \frac{k}{j-k} \rceil$, we have that $\gamma_i = \gamma_{2i}$, since $i \geqslant k$ and $2i - i = i$ is a multiple of the period $j - k$. Besides, the sequence $k, k+1, \ldots, k + (j - k - 1)$ contains a multiple of $j - k$, so that $i \leqslant j - 1$ (we can also deduce it from the formula above for i).

4. Show that if F is a random function, then the average execution time of the algorithm is in $O(p^{1/2})$ multiplications in $\mathbb{G}$.

☞ If $H : \mathbb{G} \to \mathbb{G}$ is a random function, according to the birthday paradox, the expected number of elements of the sequence (γ_i) needed to obtain two identical values is approximately $\sqrt{\pi p/2}$. Since every iteration of the **while** loop uses a constant number of multiplications in $\mathbb{G}$, the result follows.