3.1 (簡單來說，這題只是要證「如果這條式子成立則可以以線性時間內推導至另一條式子」)

要先說明，以下四條都是「不可能」達成的難題，這題只是要證彼此的難度等價

↳ (1) 可到 (2)，(2) 也可到 (1)

給定

(1) know g^α, g^β, we can compute $g^{\alpha\beta}$

(2) know g^α, we can compute g^{α^2}

(3) know g^α & $\alpha \neq 0$, we can compute $g^{\frac{1}{\alpha}}$

(4) know g^α, g^β with $\beta \neq 0$, we can compute $g^{\frac{\alpha}{\beta}}$

(1):

We have an algorithm $f_{12}(g^\alpha, g^\beta) = g^{\alpha\beta}$

(1) ⇒ (2) $f_{12}(g^\alpha, g^\alpha) = g^{\alpha^2}$ ※

(2) ⇒ (1)

① also, $f_{12}(g^\beta, g^\beta) = g^{\beta^2}$

② $g^\alpha \times g^\beta = g^{\alpha+\beta} \Rightarrow f_{12}(g^{\alpha+\beta}, g^{\alpha+\beta}) = g^{(\alpha+\beta)^2}$

$= g^{\alpha^2+\beta^2+2\alpha\beta}$ ---- $\boxed{1}$

③ $g^{\alpha^2} \times g^{\beta^2} = g^{\alpha^2+\beta^2}$ ... $\boxed{2}$

$\boxed{1} \div \boxed{2} \Rightarrow g^{2\alpha\beta} \Rightarrow$ 開根號 $\Rightarrow g^{\alpha\beta}$ #

(2)

$f_{23}(g, g^{\alpha}) = g^{\alpha^2}$

(2) ⇒ (3) $f_{23}(g^{\alpha}, g) = f_2(g^{\alpha}, (g^{\alpha})^{\frac{1}{\alpha}}) = (g^{\alpha})^{(\frac{1}{\alpha})^2} = g^{\frac{1}{\alpha}}$ #

(3) ⇒ (2) $f_{32}(g, g^{\alpha}) = g^{\frac{1}{\alpha}} = g^{\alpha^{-1}}$

$f_{32}(g^{\alpha}, g) = f_3(g^{\alpha}, (g^{\alpha})^{\frac{1}{\alpha}}) = (g^{\alpha})^{(\frac{1}{\alpha})^{-1}} = (g^{\alpha})^{\alpha} = g^{\alpha^2}$ #

(1) ⇒ (4)

$f_{14}(g, g^{\alpha}, g^{\beta}) = g^{\alpha\beta}$

$f_{14}(g^{\beta}, g, g^{\alpha}) = f_{14}(g^{\beta}, (g^{\beta})^{\frac{1}{\beta}}, (g^{\beta})^{\frac{\alpha}{\beta}})$

$= (g^{\beta})^{\frac{1}{\beta} \times \frac{\alpha}{\beta}} = g^{\frac{\alpha}{\beta}}$ #

(4) ⇒ (1)

$f_{41}(g, g^{\alpha}, g^{\beta}) = g^{\frac{\alpha}{\beta}}$

$f_{41}(g, g, g^{\alpha}) = g^{\frac{1}{\alpha}}$

$f_{41}(g, g^{\beta}, g^{\frac{1}{\alpha}}) = g^{\alpha\beta}$ #

3.2

private key c, public key $G \in G_n$, $Q_{CA} = cG$

Alice		CA
$\alpha \xleftarrow{\#} \mathbb{Z}$	$\xrightarrow{\alpha G}$	
		$k \xleftarrow{\#} \mathbb{Z}$
		compute kG
		compute $\gamma = \alpha G + kG$
		compute $Cert = Enc(\gamma, ID_A)$
		compute $e = H(Cert)$
	$\xleftarrow{(Cert,\ s)}$	compute $s = ek + c$ ---(1)

Compute $e' = H(Cert)$

Get private key

$\Rightarrow a = e'\alpha + s$ ---(2)

derives $\gamma' = Dec(Cert)$

public key

$\Rightarrow Q_A = e'\gamma' + Q_{CA}$ ---(3)

1.

from (1) $sG = ekG + cG$

from (2)
$$
\begin{aligned}
aG &= e'\alpha G + sG \\
&= e'(\gamma - kG) + ekG + cG \\
&= e'\gamma - e'kG + ekG + cG \\
&= e'\gamma + cG \\
&= e'\gamma' + Q_{CA} \\
&= Q_A \quad \text{---(3)}
\end{aligned}
$$

※ Due to $H(\cdot)$ is collision resistant

$\Rightarrow Cert = E(\gamma, ID_A)$

$\gamma' = \gamma = D(Cert)$

※

(2)

assume we can compute a valid s' without secret c

Alice will receive $(Cert, s')$, and she will compute

$a' = e'\alpha + s' = e\alpha + s'$ due to $H(\cdot)$ is collision resistant

proof: $s'G$ is independent with c

$$a'G = e\alpha G + s'G$$
$$= e(\gamma - kG) + s'G$$
$$= e\gamma' - ekG + s'G = aG = Q_A \quad \text{if } s' \text{ is valid}$$
$$\hookrightarrow e\gamma' + G(s' - ek) = aG$$
$$\Rightarrow G(s' - ek) = aG - e\gamma' = cG$$
$$\Rightarrow s' - ek = c$$
$$\Rightarrow s' = ek + c = s \quad \#$$

if a third parity wants to pass the verification without the secret c, he/she should generate a s' satisfied $s' = s$. It is computationally infeasible. #