Ex 1:
1. check $e(H(m), y) = e(0, g)$
2. assume there is such an adversary $A$
   - $A$ obtains access to public key, $OH$ (programmable random oracle), and $O_s$ (signing oracle)
   - build COH adv:
     - obtain $A=g^a$, $B=g^b$, goal: compute $g^{ab}$ using $A$
     - set public key $y = A$ and seed to $U$
     - draw some $i^*=\{1, Q\}$, where $Q$ is an upper bound on $\#OH$-queries
     - simulate $OH$:
       - $OH(q; i)$:
         - if $i=i^*$: output $B$
         - else $g^{i; r}$ for $r \leftarrow \mathbb{Z}_p$
           (Note: distribution is random, $i^*$ is hidden from $A$)
     - simulate $O_G(m)$:
       - if $m = q_i$: abort
       - else output $A^j$ where $m_i$ is the $j$-th oracle query
       - at end, if no abort, output $\overline{A}$, $\overline{m}$ for unquazed $m$
       - and $e(H(m), A) = e(\overline{A}, \overline{m})$
         $\Rightarrow \overline{A} = H(m)$
       - if $\overline{A}$ is the $i^*$-th $OH$ query, then $H(m) = B$
       - else output COH solution $\overline{A}$
     - as $\overline{A}$ hidden from $A$, prob. of that is $\frac{1}{Q}$
       (Note that in that case $\overline{m}$ never queried to $O_G$ and we never abort)
     
   with prob. $\frac{1}{Q}$ we can break COH, when $E$ is success prob. of $A$
   as $Q$ is polynomial bounded, $E$ has to be "tiny", as we assume COH is hard
3. signer
   - $y, g, x, t, y^g$

   $$c = H(m) \cdot g^r \text{ for random } r$$

   $$d = c^x \quad \Rightarrow \quad o = d \cdot y^r$$

   $$\text{then: } o = c^x \cdot g^{yr} = H(m)^x \cdot g^{x \cdot yr} = H(m)^x$$

4. first message is uniformly random in $G$, and $o$ is uniquely determined by $y$ and $H(m)$ — both cases identically distributed
5. similar to 2. but:
   - simulate signing oracle $O_s$ via the $(\cdot)^x$ oracle
   - simulate $OH$ with the second oracle: $OH(q; i) = h_i$
     as a BLS adversary gives a $Q+1$-th $h_i^x$, it cannot exist
6. it is interactive (hut: assuming one-more COH is hard is equivalent to assume that BLS is secure)
   - strong assumption
   - but: can analyze one-more COH in the generic group model
   - for generic adversaries it is a reasonable assumption

(similar to Olog analysis of previous TD)
Ex 2.
1. similar to semantic security for encryption schemes, intuitively:
   - givenExtract oracle, should be hard to distinguish ciphertexts for unpukeed skid vs. random
   - at least: should be hard to compute skid*, even if skid; is known
     (otherwise allowing with other users might allow to compute skid's session for example)
2. \(^*\) KeyGen (\(\lambda\)):
   - run \(\mathbf{mk}, \mathbf{msk} \leftarrow \text{IBE.Setup}(\lambda)\)
   - output \(\mathbf{pk} = \mathbf{mk}, \; \mathbf{sk} = \mathbf{msk}\)

\[
\text{Sign}((\mathbf{sk}, m)) \quad \text{output} \; \sigma \leftarrow \text{IBE.Extract}((\mathbf{msk}, m))
\]

\[
\text{Verify}(\mathbf{pk}, m, \sigma_m) \quad : \quad c \leftarrow \text{IBE.Encrypt}(\mathbf{mpk}, m^*) \quad \text{for random} \; m^*
\]

\[
m' \leftarrow \text{IBE.Decrypt}(\sigma_m, c)
\]

acts as decryption key

\[
\text{check} \; m = m'
\]

Note: skid should be hard to compute (even if other skid; are known)
verification checks whether the decryption key works

3. we obtain BLS
4. CDH
5. both identically distributed
6. it can be checked that with \(\mathbf{A} = (\mathbf{m} - \mathbf{m}^*)^\mathbf{t} \), \(\mathbf{B} = \mathbf{m} \cdot \mathbf{g}^\mathbf{t} \)
   for \(\mathbf{r} \leftarrow \mathbb{Z}_p^n\), then \((\mathbf{A}, \mathbf{B})\) is a valid signature for \(\mathbf{m}\).
   (note that for \(\mathbf{m} = \mathbf{m}^*\) this is not possible but as the adversary declares \(\mathbf{m}^*\) before
   we need to sign the \(\mathbf{pk}\), this is fine)
   - with \(\mathbf{pk} = (\mathbf{A}, \mathbf{B}^\mathbf{m} \cdot \mathbf{g}^\mathbf{r})\) and the above signing mechanism,
   - we can simulate the challenge
   - it's not hard to check that a signature \(\mathbf{\sigma}^*\) for \(\mathbf{m}^*\) allows to compute \(\mathbf{g}^{\mathbf{\sigma}^*} = \mathbf{g}^{\mathbf{r}^*} / \mathbf{g}^\mathbf{r}^o\)

7. no, because the adversaries usually see the public key at the signer
   before they decide for which message to forge a signature (but depends on setting of ours)
8. given \((\mathbf{a}_1, \mathbf{a}_2) = (\mathbf{sk} \cdot (\mathbf{m} \cdot \mathbf{h}^n)^\mathbf{t}, \mathbf{g}^\mathbf{r})\), then
   \((\mathbf{a}_1', \mathbf{a}_2') = (\mathbf{a}_1 \cdot (\mathbf{m} \cdot \mathbf{h}^n)^{\mathbf{\sigma}^*} \mathbf{a}_2 \mathbf{g}^\mathbf{r}^o)\) is a signature with randomness \(\mathbf{r}' = \mathbf{r} + \mathbf{r}^o\)
9. sign H(m) instead of m
   (private oracle by guessing which oracle query corresponds to the forgery, similar to in Ex 1a)

Ex 3:
1. \(5^t \mathbf{c} = (-5^t \mathbf{A})(\mathbf{t} \mathbf{A} + \mathbf{e}^t) \cdot \mathbf{r} + (5^t \mathbf{A}) \cdot \mathbf{e}^t \cdot \mathbf{r} + (\mathbf{\lambda}_{1n}^2 + \mathbf{\mu})
   \quad \times \mathbf{e}^t \cdot \mathbf{r} + (\mathbf{\lambda}_{1n}^2 + \mathbf{\mu})
   \quad \text{close to } 0 \; \mathbf{A} \quad \mathbf{\mu} = 0
   \quad \text{close to } \mathbf{L}_{1n} \; \mathbf{A} \quad \mathbf{\mu} = 1

2. \(5^t \mathbf{A} + \mathbf{e}^t\) looks random under \(\mathbb{WE}\), thus hint yields that \(\text{Encrypt}(\mathbf{u})\) is uniformly distr.
3. simple calculation (not: error increases slightly)
4. no, a non-small term appears in the product
5. add \(\text{Enc}(\mathbf{sk})\) into \(\mathbf{pk}\), add/multiply once, then decrypt homomorphically using \(\text{Enc}(\mathbf{sk})\)
   - yields "almost fresh" ciphertext (which allows for at least one more add/multiply)