1. ECB: $m_i = F_k^{-1}(C_i)$

CBC: $m_i = F_k^{-1}(C_i) \oplus C_{i-1}$

OFB: define $y_0 := IV$, $y_i = F_k(y_{i-1})$

$m_i = C_i \oplus y_{i-1}$

CTR: $m_i = F_k(ctr+i) \oplus C_i$

2. Assume the block length is n and A will use messages with length n

A queries the oracle with an arbitrary message m and get (IV, C)

A chooses two messages m_0 and m_1 such that

$m_0 = m \oplus IV \oplus (IV+1)$ and $m_1 \neq m_0$

A gives m_0 and m_1 to Π and get $(IV+1, C')$

If m_0 is encrypted,

$C' = F_k(m_0 \oplus (IV+1)) = F_k(m \oplus IV \oplus (IV+1) \oplus (IV+1))$

$= F_k(m \oplus IV) = C$

If m_1 is encrypted,

$C' \neq C$ unless $F_k(m_1 \oplus (IV+1)) = F_k(m \oplus IV)$

which happens with a negligible probability

Thus, A output 0 if $C = C'$, 1 if $C \neq C'$

A can win with a non-negligible probability, so the scheme is not secure.

3. According to the decryption logic for CBC, OFB, CTR, only CBC requires another cipherblock to decrypt a cipher block. Thus, in OFB and CTR, a single-bit error in the cipherblock has no effect on decrypting other blocks.
If a cipherblock c_k has a single-bit error, CBC will fail to decrypt 2 blocks c_k and c_{k+1}, OFB and CTR will only fail to decrypt 1 block c_k.

4.
```
byte recover[m]
for (int i=0; i<m; i++) {
    int time[256]
    for (byte b=0; b<256; b++) {
        recover[i]=b
        Vrfy_k(m, recover)
        time[b] = time to execute Vrfy
    }
    recover[i]= argmax_b (recover[b])
}
```
For the outer loop, the first iteration takes $256T$ ms, the second iteration takes $256 \times 2T$ ms, ... the mth iteration takes $256 \times mT$ seconds.
In total, it takes $256 \times (T+2T+\cdots+mT)$

$$= \frac{m(1+m)}{2} \times 256T$$

$$= 128\,m(1+m)T \text{ ms}$$