4 Cryptography (mk428)

(*a*) You have intercepted a ciphertext c from the communications of an international criminal gang, and you need to decrypt it. By decompiling the gang's messaging app you learnt that c is encrypted with AES-128 in CBC mode; the plaintext is padded to a multiple of 128 bits by setting the last plaintext byte to be the padding length in bits (encoded as an 8-bit binary number), and setting the remaining bits between the end of the message and the final length byte to zero.

Moreover, you know that the gang operates a server that is reachable over the Internet; this server internally decrypts any ciphertext you send it, and always replies with "ok" (regardless of whether the decrypted data makes sense or not). You cannot break into the server, but you notice one detail: if the decrypted message has correctly formatted padding, the reply is slightly slower than if it has incorrect padding. Presumably this is because the server spends some time storing correctly formatted messages, while malformed messages are quickly discarded without being stored.

Show that, by repeatedly sending messages to the server, you can recover the entire plaintext from c. Explain your technique in detail. [8 marks]

(*b*) The gang figures out that you are decrypting their messages. They decide to continue using AES-128-CBC, but in order to prevent the attack from part (*a*), they add a check to their encryption scheme so that the server rejects any message where the ciphertext has been manipulated. Explain how to securely compute such a check using the SHA-256 hash function. [3 marks]

(*c*) As part of the new check from part (*b*), the server uses the following pseudocode:

```
// tagInMessage and correctTag are byte arrays of equal length
function checkIsMessageOk(tagInMessage, correctTag) {
    for (i = 0 to correctTag.lengthInBytes - 1) {
        if (tagInMessage[i] != correctTag[i]) {
            send "rejected" reply
            return
        }
    }
    send "ok" reply
}
```

Explain the problem with this code, and show how this problem may once again allow you to recover the entire plaintext of an encrypted message. [3 marks]

(*d*) Prove that if a hash function $H(x)$ is collision resistant, then $H(H(x))$ is collision resistant as well. [6 marks]