Homework 4

1 Problem 1 (25 points)

(a) Suppose Alice and Bob have established a secret key k_{AB} for a CPA-secure encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$, and that Alice and Charlie have established a secret key k_{AC}. Charlie does not know k_{AB} and Bob does not know k_{AC}.

Alice wishes to encrypt an n-bit message m such that it can be decrypted by either Bob or Charlie. However, consider m to be a very large message, so Alice wishes to minimize the amount of data she sends. Devise a way for her to encrypt m using $n + \mathsf{poly}(\lambda)$ bits.

(b) Again, suppose Alice has established secret keys k_{AB}, k_{AC} with Bob and Charlie, respectively.

Now, Alice wishes to encrypt an n-bit message m such that it can be decrypted by Bob and Charlie if they cooperate, but cannot be decrypted by Bob or Charlie individually. Bob and Charlie share a private channel over which they can communicate.

Devise a solution to this problem. Your solution should not compromise k_{AB} or k_{AC}: after decrypting Alice's message, Charlie should still not know k_{AB} (so Alice and Bob can still communicate while excluding Charlie) and Bob should still not know k_{AC}. Again, you should ensure that the ciphertext size is $n + \mathsf{poly}(\lambda)$.

(c) Now suppose Alice has established secret key k_{AB}, k_{AC}, k_{AD} with Bob, Charlie, and Donald, respectively. Alice wishes to encrypt an n-bit message m such that it can be decrypted by any two of **{Bob,Charlie,Donald}**, but not by any one of them individually.

Devise a solution to this problem that does not compromise any of the keys. Again, the ciphertext size should be $n + \mathsf{poly}(\lambda)$.

(d) More generally, Alice has esablished secret keys k_{Ai} for $i = 1, \ldots, \ell$ with ℓ different individuals. Alice wishes to encrypt m such that any subset of t of the individuals can decrypt, but any $t-1$ of them cannot. Devise a solution to this problem.

(e) Define the overhead of your solution as the length of the ciphertext minus the length of the message. How does the overhead in your solution scale with ℓ and t?

[We will later see a much more efficient way to solve this problem]

2 Problem 2 (15 Points)

Often when using block ciphers for encryption, messages need to be padded to a multiple of the block size. For each of the following padding schemes, decide if the padding is reversible: that is, for any message, after padding to a multiple of the block length, it is possible to recover the message again. If the padding is reversible, explain how to recover the message and why recovery is guaranteed to work. If not, explain how it fails.

(a) **Null Padding**: Append 0's to the message until it is a multiple of the block length

(b) **Bit Padding:** Let N be the number of bits necessary to pad to a multiple of the block length. To ensure that $N > 0$, if the message is already a multiple of the block length, let N be the block length. Append 10^{N-1} — that is, a 1 followed by $N - 1$ 0's — to the message.

(c) **PKCS7 Padding:** Assume the message is an integral number of bytes. Let N be the number of *bytes* necessary to pad to a multiple of the block length. To ensure that $N > 0$, if the message is already a multiple of the block length, let N be the block length (in bytes). Now pad with N bytes, each byte set to the value N. For example, if $N = 3$, append 3 bytes to the message, each byte set to 00000011.

(d) PKCS7 padding, except that if the message is already a multiple of the block length, do not add any padding.

3 Problem 3 (40 Points)

Consider CBC mode encryption, using PKCS7 padding from Problem 2c. We now will see a potential attack on CBC mode encryption using this padding. Suppose Alice is sending messages to a server, encrypted with CBC mode using PKCS7 padding. The server decrypts the CBC encryption, and then checks that the padding is correct: it verifies that there is some $N > 0$ such that the last N bytes of the plaintext (after decrypting, but before stripping the padding) are set to N. If correct, the server does

nothing. If not, the server sends a "reject" message back to Alice indicated that the message was not well-formed.

Suppose an adversary Eve sits between Alice and the server. Eve can intercept messages from Alice, as well as send messages to the server, and read the response from the server. However, Eve does not know the secret key ALice and the server are using to communicate. Here is a sketch of Eve's attack:

- Eve intercepts a ciphertext c from Alice.
- Eve has a guess g for the last byte in the last full block of plaintext. In other words, if the message length is $\ell s + t$ bytes, where ℓ is the number of bytes per block and $t < \ell$, then Eve has a guess g for byte ℓs.
- Eve first strips off the last block of c, obtaining c'
- Next, Eve changes some portion of c' obtaining c''.
- Eve sends c'' to the server, and based on the server's response, learns whether or not g was a correct guess.

Answer the following:

(a) Fill in the details of the attack, namely decide how Eve computes c''. [Hint: you only need to modify one byte of c']

(b) Extend the attack to actually learn byte ℓs given no other information about the message, by making multiple queries to the server. [Hint: there are only 256 possible values for that byte]

(c) Under what circumstances will the attack in (a),(b) not be guaranteed to give the right answer? Can Eve tell if this situation arises?

(d) Explain how to still recover the byte, even if the attack in (a),(b) was inconclusive.

(e) Explain how to extend the attack to recover the entire message. This includes learning the first ℓs bytes, as well as the last t bytes.

(f) But wait! We said that CBC encryption was CPA secure, and yet we just showed how to recover the entire message. Explain why this is not a contradiction.

4 Problem 4 (20 Points)

For each of the following block cipher modes, consider the following. Alice and Bob share a secret key k for a block cipher with block length ℓ bites. Alice has an s-block message m, which she encrypts, sending the ciphertext c to Bob. Now, suppose in transmission, a single bit at position i is flipped, resulting in Bob receiving a ciphertext c'. Bob decrypts, obtaining a message m'. How many bits or blocks of m' have been corrupted by the single bit flip in the ciphertext? For modes with an IV, assume the bit was not a part of the IV.

(a) ECB Mode

(b) Counter Mode

(c) CBC Mode

(d) OFB Mode

(e) CFB Mode