Massachusetts Institute of Technology
6.5610: Applied Cryptography — February 19, 2025
Henry Corrigan-Gibbs and Yael Tauman Kalai — **Due:** February 28, 2025 at 4:59pm

Problem Set 2

Please submit your problem set, in PDF format, on Gradescope. *Each problem should be in a separate page.*

You are to work on this problem set in groups. For problem sets 1, 2, and 3, we will randomly assign the groups for the problem set. After problem set 3, you are to work on the following problem sets with groups of your choosing of size three or four. If you need help finding a group, try posting on Piazza. See the course website for our policy on collaboration. Each group member must independently write up and submit their own solutions.

Homework must be typeset in LaTeXand submitted electronically! Each problem answer must be provided as a separate page. Mark the top of each page with your group member names, the course number (6.5610), the problem set number and question, and the date. We have provided a template for LaTeX on the course website (see the *Psets* tab at the top of the page).

Problem 2-1. LWE

Recall that the LWE assumption, with respect to parameters n, m, q, χ asserts that

$$(\mathbf{A}, \mathbf{sA} + \mathbf{e}) \approx (\mathbf{A}, \mathbf{u})$$

where $\mathbf{A} \xleftarrow{\text{R}} \mathsf{Z}_q^{n\times m}$, $\mathbf{s} \xleftarrow{\text{R}} \mathsf{Z}_q^n$, $\mathbf{e} \leftarrow \chi^m$, and $\mathbf{u} \leftarrow \mathsf{Z}_q^m$.

For each of the following parts add a short explanation for your answer.

(a) Consider a variant of the LWE assumption where the first $n/2$ rows of $\mathbf{A}$ are set to 0. Is this assumption broken or does it remain secure under LWE (with some parameters)?

(b) Consider a variant of the LWE assumption where the first column of $\mathbf{A}$ is set to 0. Is this assumption broken or does it remain secure under LWE (with some parameters)?

(c) Consider a variant of the LWE assumption where $\mathbf{A}$ is chosen random in $\mathbb{Z}_q^{n\times m}$ subject to the last column being the sum of the first two columns. Is this assumption broken or does it remain secure under LWE (with some parameters)?

(d) Consider a variant of the LWE assumption where $\mathbf{s}$ is chosen randomly in $\mathbb{Z}_q^n$ subject to $s_n = \sum_{i=1}^{n-1} s_i$. Assume that $q = \mathsf{poly}(n)$. Is this assumption broken or does it remain secure under LWE (with some parameters)?

(e) **Extra credit.** The same problem as part (d) except that the modulus is $q = 2^n$.

Problem 2-2. Message Authentication Codes

Message Authentication Codes (MACs) are used to verify the integrity of messages that are sent (i.e. MACs prevent an adversary from tampering with a message). A MAC is associated with a message space $\mathcal{M} = \{\mathcal{M}_\lambda\}_{\lambda\in\mathbb{N}}$. It takes as input a secret key $k \leftarrow \{0,1\}^\lambda$ and a message $m \in \mathcal{M}_\lambda$, and outputs a *tag*, which should be thought of as a "proof" of the authenticity of m. Formally, a MAC is said to be secure if any poly-time adversary $\mathcal{A}$ wins the following game with negligible probability:

- Challenger samples MAC key $k \xleftarrow{R} \{0,1\}^\lambda$.
- The following is repeated polynomially many times:
  - Adversary sends any message $m_i \in \mathcal{M}$ to the challenger
  - Challenger responds with $t_i = MAC(k, m_i)$
- Adversary sends the challenger a message-tag pair (m^*, t^*)

- Adversary wins if $MAC(k, m^*) = t^*$ and $m^* \notin \{m_1, m_2, ...\}$. That is, the adversary wins if it can construct a valid message-tag pair for a message that has not already been sent.

We often use PRFs to construct MACs (even though MACs are not required to be pseudo-random). This can be done by taking any PRF $F : \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}^\lambda$, and setting the MAC to be:

$$MAC(k, m) = F(k, m).$$

Notice that this is a secure MAC for the message space $\mathcal{M}_\lambda = \{0,1\}^\lambda$. In what follows, we examine a few possible MAC constructions that use the PRF F to MAC longer messages. For simplicity, think of messages whose length is a multiple of λ. For each of the following constructions, state whether it is secure, and if it is not secure then provide an attack. In what follows $m = m_1||m_2||...||m_n \in \{0,1\}^{n\cdot\lambda}$ and each $m_i \in \{0,1\}^\lambda$.

(a) $MAC(k, m) = (F(k, m_1), F(k, m_2), ..., F(k, m_n))$.

(b) $MAC(k, m_1||m_2||...||m_n) = F(k, m_1) \oplus F(k, m_2) \oplus \ldots \oplus F(k, m_n)$, which means that we apply the PRF on every block and xor the results together.

(c) $MAC(k, m) = F(MAC(k, m_1||m_2||...||m_{n-1}), m_n)$ and $MAC(k, m_1) = F(k, m_1)$. This is a recursive function where the result of the previous block's MAC serves as the key for the next block.

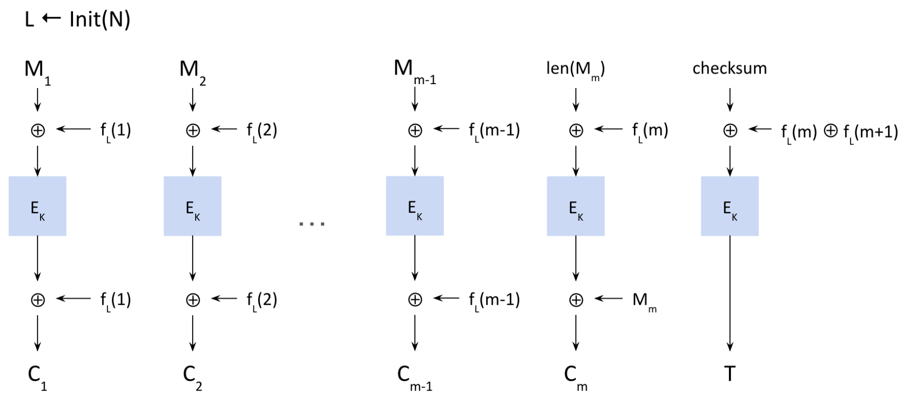
Problem 2-3. OCB Mode

We want to have encryption schemes that achieve both *confidentiality* and *integrity*. This may be achieved by combining an encryption scheme that is CPA secure, as described in lecture 3, with a MAC that satisfies EUF-CMA. In this problem, we explore a block cipher-based mode of operation which achieves both with some additional desirable properties. Offset codebook mode (OCB mode) provides *authenticated encryption*, meaning that the scheme achieves both confidentiality and integrity.

An older version of the OCB encryption algorithm $\mathsf{Encrypt}(N, M)$ is described by the diagram below. In this construction, we assume that both every message block and tag have length n for simplicity.

- N is a nonce, and M is the message.
- $E_k : \{0,1\}^n \to \{0,1\}^n$ is a block cipher (e.g. AES-128).
- $\mathsf{len}(S) : \{0,1\}^* \to \{0,1\}^n$ is a function which outputs an n-bit encoding of the length of input bitstring S.
- $\mathsf{checksum} = M_1 \oplus M_2 \oplus \cdots \oplus M_m$.
- $f_L(i)$ is a function which outputs an n-bit string. We can ignore the details of f_L for this problem.

For now, ignore the implementation of $\mathsf{Init}(N)$ and consider L to be a constant.



(a) Describe the decryption algorithm $\mathsf{Decrypt}(N, C, T)$ for OCB mode. To decrypt correctly, it must be that both the ciphertext $C = C_1 \| \ldots \| C_m$ decrypts to the correct message and T is a valid tag.

(b) In lecture, we saw two other methods for encryption. *(1) Counter mode to encrypt with a PRF* and *(2) CBC mode to encrypt with a PRP*. For each method, name one advantage of using OCB mode in comparison to it.

We now explore a *message forgery attack* on this construction. After seeing one encryption of a chosen message, an adversary is able to construct a valid tag on a different ciphertext. In the first step, the adversary asks for an encryption of (N, M) where N is any nonce and $M \in \{0,1\}^{2n}$ defined by $M = M_1 \| M_2$ where $M_1 = \mathsf{len}(0^n)$ and $M_2 \in \{0,1\}^n$.

This encryption will give ciphertext $C = C_1 \| C_2$ where

$$C_1 = f_L(1) \oplus E_k(f_L(1) \oplus \mathsf{len}(0^n))$$
$$C_2 = M_2 \oplus E_k(f_L(2) \oplus \mathsf{len}(0^n))$$

(c) Complete the attack. That is, given the above C, construct ciphertext C' and tag T' such that the decryption algorithm will accept $(N' = N, C', T')$. *Hint: C' will only be n bits long.*

(d) We now describe how L is constructed below. For the rest of this problem, suppose N is a counter. That is, every time we call the encryption function, we increment $N = 1, 2, \ldots$.

$\mathsf{Init}(N) \to L$:

- $\mathsf{Bottom} \leftarrow N[-6:]$
- $\mathsf{Top} \leftarrow N[:-6] \|$constant padding
- $K_{\text{top}} \leftarrow E_k(\mathsf{Top})$
- $\mathsf{Stretch} \leftarrow K_{\text{top}} \| (K_{\text{top}} \oplus (K_{\text{top}} << 8))$
- $L \leftarrow \mathsf{Stretch} << \mathsf{Bottom}[0:128]$

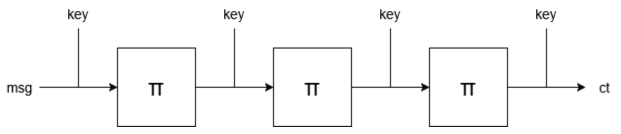
After how many calls does Bottom need to be recomputed? Top? K_{top}?

(e) Using the previous part, explain why the amortized cost to compute Init is low.

Problem 2-4. Slide attack

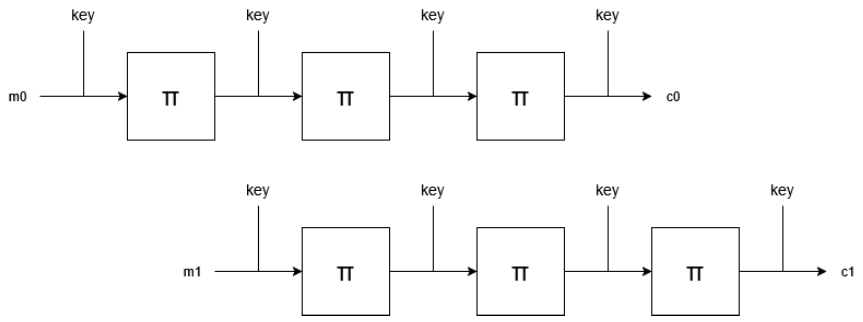
In the first pset, we see how one can break a single-round cipher with a birthday attack. What if we increase the number of rounds? The encryption is much more complex, and the birthday attack no longer works, we must be secure, right? The answer is "No". In this problem, you are going to break a single-key 300-round Even-Mansour encryption scheme.

Let $\pi\colon \{0,1\}^n \to \{0,1\}^n$ be a random permutation and let $k \in \{0,1\}^n$ be the key. Define $f(x) = \pi(x) \oplus k$, the encryption is:



$$\mathsf{Enc}(m) = f^{300}(m \oplus k)$$

where f^i means applying the function i times. In total, we apply the permutation π 300 times and xor with the key 301 times. The slide attack takes advantage of the repeating pattern of the encryption scheme: if two messages are "one step" away, then the ciphertexts are also "one step" away.



From the figure, we can see that if $m_1 = \pi(m_0 \oplus k)$, then $c_1 = \pi(c_0) \oplus k$. We call such a pair (m_0, m_1) a sliding pair.

(a) Find two functions f and g such that $f(m_0, c_0) = g(m_1, c_1)$ for any sliding pair. Similarly to pset 1, both functions can include π (and π^{-1}) but not k.

(b) Sample two random functions $f : [N] \to [N]$, $g : [N] \to [N]$, and t random inputs $m_1, m_2, \ldots, m_t$. Prove that if $t = \Theta(\sqrt{N})$, there is a constant probability that $f(m_i) = g(m_j)$ for some i, j. (You don't have to find the best constant.)

Given the information above, we can find a sliding pair with sample size as large as that of a normal birthday attack. Now we outline the attack:

1.Collect many pairs of $(m_i, c_i = \mathsf{Enc}(m_i))$.

2.Find i, j such that (m_i, m_j) is a sliding pair.

3.Recover the secret key from (m_i, m_j).

(c) On Piazza, you can find a zip file `pset2.zip` that contains two files:

- `main.py` generates a random key and 2^{20} ciphertexts. It is for you to understand how the data was generated.

- `data.txt` has the ciphertexts generated by `main.py` and it is not human-readable. You should use the helper function to read the data.
- `lib.py` has the implementation of encryption and a helper function to read the data.

The goal is to find the secret key. Note that using the `read_data` in `lib.py`, you get an array of size 2^{20}, and the i-th element is the encryption of the number i (0-indexed). Put your secret key as the answer, and submit the code to Gradescope.