# COMP 4820

# Project

# POODLE
# (Padding Oracle On Downgraded Legacy Encryption)

## Demo of the Attack
## Link: https://youtu.be/NDghAnay3jA

Ujjwal Singla
MD Sakif Al Mohaimen
Sarwar Ahsan
Tanaya Siddiqui
Arnav Verma

**Padding Oracle Attack Report**

## Abstract

This document reports on a padding oracle attack simulation conducted on a web application's encryption system. The attack exploits error messages from the encryption service to decrypt cookies without the encryption key. The report aims to detail the vulnerability, demonstrate the attack, and suggest mitigation strategies.

## Introduction

The padding oracle attack is a well-known exploit that takes advantage of the padding validation process in block cipher encryption algorithms. It allows an attacker to decrypt data without the key by observing the response to manipulated ciphertext. This report demonstrates such an attack on an Express.js web application that uses AES-256-CBC encryption for cookie data.

## System Overview

The application uses `crypto` module from Node.js to encrypt and decrypt user data, specifically the 'username' field stored in cookies. The encryption system uses AES-256-CBC mode with a secret key and initialization vector (IV). The encryption process is designed to secure user sessions and sensitive information against unauthorized access.

## Vulnerability Assessment

The application's decryption function returns specific errors when decryption fails due to incorrect padding, making it susceptible to a padding oracle attack. This behavior provides attackers with information about the padding validity of a given ciphertext, leading to potential data exposure.

## Attack Scenario

An attacker intercepts the encrypted `cookieEncrypted` and attempts various modifications. The application's response to each modification—either 'incorrect padding' or 'decryption failed'—gives the attacker clues about the padding's correctness. Over time, this information can be used to decrypt the entire cookie without prior knowledge of the encryption key.

## Proof of Concept

The proof of concept was conducted by intercepting the `cookieEncrypted` after a user login. By sending different payloads and observing application responses, it was possible to decrypt the cookie contents incrementally. The response messages played a critical role in this process, allowing the padding to be guessed correctly.

## Impact Analysis

A successful padding oracle attack on this system could lead to unauthorized access to user accounts, session hijacking, and private data disclosure. This represents a significant security threat, as the system handles encryption at a foundational level.