分組名單（不足 5 個人空著就好）：

姓名	學號
陳孟楷	113550021
蔡昀呈	113550058

1. Name of the paper:
   NIST Special Publication 800-38D: Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC.

2. Summary:
   This publication defines two cryptographic modes of operation: Galois/Counter Mode (GCM) and GMAC. GCM is used for authenticated encryption with associated data, where it guarantees both the confidentiality of the data and its authenticity. GMAC, a variant of GCM, generates a message authentication code (MAC) for data that does not require encryption, solely for authentication purposes. Both GCM and GMAC operate on a symmetric key block cipher (such as AES) and involve two main functions: authenticated encryption (encrypting the data and generating an authentication tag) and authenticated decryption (decrypting the data while verifying its authenticity using the authentication tag). The paper provides detailed specifications, including the mathematical components of GCM, implementation guidelines, and considerations for the uniqueness of initialization vectors (IVs) and keys.

3. Strength(s) of the paper:

- **Thorough Specifications**: The paper provides an exhaustive description of GCM and GMAC, including detailed mathematical operations (such as GHASH and GCTR) and their implementation.
- **Clear Definitions**: It offers precise definitions and notations for every element of the algorithm, making it accessible for both cryptographers and implementers.
- **Implementation Guidelines**: The document includes practical guidelines for the proper implementation of GCM, addressing real-world issues such as the importance of key and IV uniqueness and offering considerations for cryptographic module validation.

- **Security Assurance**: The paper carefully considers security aspects, highlighting the importance of ensuring no repeated IVs for the same key, and gives a strong foundation for the use of GCM in secure environments.

4. Weakness(es) of the paper

- **Complexity for Beginners**: The paper is highly technical and assumes a background in cryptography and block ciphers, making it difficult for a general audience or those new to encryption technologies to fully grasp.
- **Limited Discussion on Performance**: While the paper focuses on the cryptographic aspects of GCM, it could provide more insight into performance considerations, particularly when dealing with large datasets or high-speed applications.
- **Lack of Discussion on Real-World Failures**: Although the paper mentions the theoretical vulnerabilities of GCM (such as IV reuse), it does not delve deeply into real-world failures or practical exploits that have occurred with implementations, leaving a gap in applied security analysis.

5. Your own reflection, which can include but not limited to:

A. **What did you learn from this paper?**

The primary takeaway from this paper is the importance of both encryption and authentication in modern cryptographic systems. GCM not only ensures the confidentiality of the data but also its integrity and authenticity. I learned how the design of GCM—using the counter mode of encryption combined with a universal hash function for authentication—provides both high efficiency and robust security properties.

B. **How would you improve or extend the work if you were the author?**
If I were the author, I would focus on:

- **Real-World Case Studies**: Include examples of GCM implementations in real-world systems, discussing potential pitfalls, failures, and solutions.
- **Performance Optimization**: Offer more details about optimizing GCM for high-performance environments, such as using hardware accelerators or parallelizing certain operations.
- **Expand on Security Threats**: While the paper outlines the theoretical risks (such as IV reuse), I would include more detailed analyses of known attacks or practical vulnerabilities found in deployed systems.

C. **What are the unsolved questions that you want to investigate?**

- How can GCM be made more resistant to specific attacks, especially in high-risk scenarios like cloud storage or blockchain applications?
- Are there more efficient variants of GCM that can be deployed without compromising security, particularly for large-scale systems?

D. **What are the broader impacts of this proposed technology?**

GCM and GMAC are essential building blocks in the field of secure communications, particularly in systems where both data confidentiality and integrity are critical. The ability to authenticate both the data and additional information (like headers) opens up many applications, from securing internet traffic (e.g., HTTPS) to encrypted messaging systems. Their efficiency also enables their use in resource-constrained environments like IoT devices, ensuring secure communication even with limited processing power.

E. **Else?**

The focus on IV uniqueness and key management is critical for the practical application of GCM, and ensuring that these are adhered to will prevent many common vulnerabilities in encryption systems.

6. Realization of a technical specification or algorithm as a program:

The Python code provided implements the Galois/Counter Mode (GCM) encryption and decryption using the PyCryptodome library, a popular cryptography library for Python. The main objective of the code is to showcase how to encrypt and decrypt data securely using AES in GCM mode, which provides both confidentiality and authenticity.

The gcm_encrypt function performs encryption using the AES algorithm in GCM mode. It generates a random 12-byte initialization vector (IV), encrypts the plaintext, and returns the ciphertext along with an authentication tag. The authentication tag is a cryptographic checksum that ensures the integrity of both the ciphertext and any additional authenticated data (AAD).

The gcm_decrypt function decrypts the ciphertext using the same AES key and IV. It verifies the authenticity of the ciphertext by comparing the provided authentication tag with one computed during the decryption process. If the tags

match, the data is considered authentic, and the plaintext is returned; otherwise, an error message is displayed.

This implementation highlights important aspects of GCM, such as IV generation, encryption, authentication, and decryption. It demonstrates the cryptographic process in a simplified form, making it a valuable reference for understanding AES-GCM in practice. However, for production environments, additional considerations like secure key management and strict IV uniqueness should be applied to ensure the highest level of security.

```
from Crypto.Cipher import AES
from Crypto.Random import get_random_bytes
from Crypto.Util import Counter
import hashlib

# GCM 加密函數
def gcm_encrypt(key, plaintext, associated_data):
    # 生成隨機的初始化向量（IV），根據規範建議使用 96 位
    iv = get_random_bytes(12)  # GCM 通常使用 96 位 IV

    # 創建 AES-GCM cipher
    cipher = AES.new(key, AES.MODE_GCM, nonce=iv)

    # 對數據進行加密，並生成認證標籤（authentication tag）
    ciphertext, tag = cipher.encrypt_and_digest(plaintext)

    # 返回加密的密文、認證標籤和 IV
    return iv, ciphertext, tag

# GCM 解密函數
def gcm_decrypt(key, iv, ciphertext, tag, associated_data):
    # 創建 AES-GCM cipher
    cipher = AES.new(key, AES.MODE_GCM, nonce=iv)

    # 嘗試解密並驗證標籤
    try:
        plaintext = cipher.decrypt_and_verify(ciphertext, tag)
        return plaintext
    except ValueError:
```

```
        print("解密失敗或認證標籤錯誤!")
        return None

# 測試程式
if __name__ == '__main__':
    # 密鑰（在實際應用中，密鑰應該安全生成並管理）
    key = get_random_bytes(16)  # AES-128 密鑰

    # 要加密的明文
    plaintext = b"這是一個測試的明文，GCM 模式測試!"

    # 附加的認證數據（例如，協議頭等不需要加密的數據）
    associated_data = b"附加數據，如協議頭"

    # GCM 加密
    iv, ciphertext, tag = gcm_encrypt(key, plaintext, associated_data)
    print("加密完成，IV:", iv.hex())
    print("密文:", ciphertext.hex())
    print("認證標籤:", tag.hex())

    # GCM 解密
    decrypted_text = gcm_decrypt(key, iv, ciphertext, tag,
associated_data)

    if decrypted_text:
        print("解密後的明文:", decrypted_text.decode())
```