BÁO CÁO BÀI TẬP

**Môn học: Mật mã học**

**Kỳ báo cáo: Buổi 02 (Session 02)**

**Tên chủ đề: AES**

*Ngày báo cáo: 09/04/2023*

1. **THÔNG TIN CHUNG:**

*(Liệt kê tất cả các thành viên trong nhóm)*

Lớp: NT219.N22.ATCL

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Nguyễn Trần Anh Đức | 21521964 | 21521964@gm.uit.edu.vn |
| 2 | Nguyễn Hữu Tiến | 21590479 | 21590479@gm.uit.edu.vn |
| 3 | Lê Thanh Duẩn | 19521370 | 19521370@gm.uit.edu.vn |

1. **NỘI DUNG THỰC HIỆN:[[1]](#footnote-1)**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Công việc** | **Kết quả tự đánh giá** | **Người đóng góp** |
| 1 | Điểm yếu của AES | 90% | Thanh Duẩn |
| 2 | Điểm yếu của mode CBC | 90% | Hữu Tiến |
| 3 | AES Benchmark Full Modes | 90% | Anh Đức |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
| 7 |  |  |  |
| 8 |  |  |  |

**Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.**

**1. AES Weakness :**

**- One potential weakness of AES is related to side-channel attacks.** **These attacks are not specific to AES, but they are a concern for any cryptographic system.**

**- Another potential weakness of AES is related to the key size. AES supports key sizes of 128, 192, and 256 bits.The use of shorter keys, such as 128-bit keys, may be vulnerable to attacks that exploit weaknesses in the key scheduling algorithm and quantum attack in the future.**

**- Related-Key Attacks: In some cases, it may be possible to recover the key used for AES by analyzing related keys. This type of attack is not practical in most situations but can be a concern for applications that use related keys.**

**2. Mode CBC Weakness :**

**CBC can be visualized as follows:**

**Step1: The data will be divided into small blocks, the first block will be XORed with an IV (IV is a random thing to ensure that the same input will produce many outputs, avoiding relay attacks), then the result (block-XOR-IV) will be input into a block encryption algorithm such as AES, and produce the first block ciphertext.**

**Step2: This ciphertext will then be used as an IV for the following plaintext blocks (as shown), and so on until the end.**

**Step3: Concatenate the resulting cipher blocks to create a complete ciphertext.**

**Here we can think of the XOR operation. From there, we have an idea for attacking as follows:**

**- This is not a direct decryption attack, but simply changing the ciphertext so that the original data changes by creating a key and an initialization vector, then performing CBC AES encryption and flipping the bit.**

**- The decryption part of CBC:**

**x[1] = D(y[1]) xor IV**

**x[n] = D(y[n]) xor y[n-1]**

**If we flip a bit in the (n-1)th ciphertext block, the bit in the (n)th ciphertext block will also be flipped.**

**3. Benchmark AES full modes with different input size:**

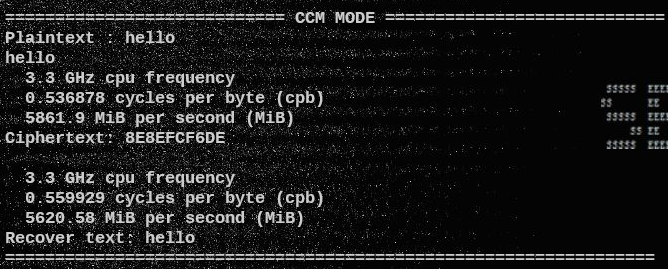
**Note : GCM did not work for some reason. I tried sample code from offical Cryptopp page and it still not working.**

**3.1 Input < 64 bit:**

****

**Text

Description automatically generated**

****

|  |  |  |
| --- | --- | --- |
| **Mode** | **Cycle per byte** | **MiB per second** |
| **ECB** | **0.40626** | **7746.57** |
| **CBC** | **1.7595** | **7783.37** |
| **OFB** | **1.92469** | **1635.13** |
| **CFB** | **4.37954** | **718.596** |
| **CTR** | **0.618515** | **5088.2** |
| **XTS** | **0.69919** | **4501.1** |
| **CCM** | **0.536878** | **5861.9** |

**3.1.1 Encryption Benchmark**

|  |  |  |
| --- | --- | --- |
| **Mode** | **Cycle per byte** | **MiB per second** |
| **ECB** | **0.425469** | **7396.84** |
| **CBC** | **0.40434** | **7783.37** |
| **OFB** | **1.84786** | **1703.12** |
| **CFB** | **4.84055** | **650.159** |
| **CTR** | **0.628119** | **5010.4** |
| **XTS** | **0.712636** | **4416.17** |
| **CCM** | **0.559929** | **5620.58** |

**3.1.2 Decryption Benchmark**

**3.2 UTF16 Input:**

****

**Text

Description automatically generated**

**A black screen with white text

Description automatically generated with low confidence**

|  |  |  |
| --- | --- | --- |
| **Mode** | **Cycle per byte** | **MiB per second** |
| **ECB** | **0.354397** | **8880.22** |
| **CBC** | **1.75566** | **1792.56** |
| **OFB** | **1.90933** | **1648.29** |
| **CFB** | **4.37954** | **718.596** |
| **CTR** | **0.484055** | **6501.59** |
| **XTS** | **0.706874** | **4452.17** |
| **CCM** | **0.486936** | **6463.12** |

**3.2.1 Encryption Benchmark**

|  |  |  |
| --- | --- | --- |
| **Mode** | **Cycle per byte** | **MiB per second** |
| **ECB** | **0.334228** | **9416.09** |
| **CBC** | **0.403379** | **7801.9** |
| **OFB** | **1.93622** | **1625.4** |
| **CFB** | **4.41028** | **713.589** |
| **CTR** | **0.497501** | **6325.87** |
| **XTS** | **0.626198** | **5025.77** |
| **CCM** | **0.486936** | **6463.12** |

**3.2.2 Decryption Benchmark**

**3.3 Input > 1Mb :**

****

**Text

Description automatically generated**

|  |  |  |
| --- | --- | --- |
| **Mode** | **Cycle per byte** | **MiB per second** |
| **ECB** | **0.358239** | **8784.99** |
| **CBC** | **1.7595** | **1788.65** |
| **OFB** | **2.12446** | **1481.37** |
| **CFB** | **4.70225** | **669.281** |
| **CTR** | **0.508066** | **6194.33** |
| **XTS** | **0.658853** | **4776.68** |
| **CCM** | **0.511907** | **6147.84** |

**3.3.1 Encryption Benchmark**

|  |  |  |
| --- | --- | --- |
| **Mode** | **Cycle per byte** | **MiB per second** |
| **ECB** | **0.339991** | **9256.5** |
| **CBC** | **0.408181** | **7710.12** |
| **OFB** | **2.02458** | **1554.46** |
| **CFB** | **4.90202** | **642.006** |
| **CTR** | **0.500382** | **6289.44** |
| **XTS** | **0.612752** | **5136.05** |
| **CCM** | **0.543601** | **5789.4** |

**3.3.2 Decryption Benchmark**

**Overview :**

**- Encryption and Decryption time do not have a remarkable differences despite of the size of the plaintext.**

**=> Small block sizes are more expensive than large block sizes**

**- ECB Mode is the fasted mode.**

**- CFB Mode is the slowest , next is OFB.**

**- Other modes speed are quite similar.**

1. Ghi nội dung công việc, các kịch bản trong bài Thực hành [↑](#footnote-ref-1)