ATS\_Crypto\_IotT\_Week1\_WS2023

**Gruppe: H**

**Teilnehmer:**

Mehmet Odabas

Baran Evsen

Omar Haj Abdulaziz

Bünyamin Berber

**Für alle Aufgaben sind die Codes in der mitgegeben Zip Datei enthalten. Es werden hier nur die benutzten Funktionen genannt und die Ausgaben gezeigt. Für beider Versionen V2 und V3 wird derselbe Code benutzt.**

**Blatt 1**

**Aufgabe 1-4**

Hardware/Arduino IDE und Bibliotheken installiert. Für NMCU V2 musste ein Treiber installiert werden.

**Aufgabe 5**

Egal welche Baudrate man benutzt hat, es gab immer wieder unvollständige Ausgaben, darum wird in unserem Code nach Serial.begin(<baudrate>), ein delay(time) gesetzt. Das Problem mit der Ausgabe gibt es dann nicht mehr.

**Aufgabe 6**

Es wurde derselbe Code von Test\_AES benutzt und die “perfRandomTextCipher” Funktion hinzugefügt. Die Durchschnittszeit ist zwischen der Version V2 und V3 identisch.

**Ausgabe NMCU V2:**

AES-128-Random-Text

Average Encryption Time: 95 microseconds

Average Decryption Time: 142 microseconds

AES-192-Random-Text

Average Encryption Time: 114 microseconds

Average Decryption Time: 171 microseconds

AES-256-Random-Text

Average Encryption Time: 133 microseconds

Average Decryption Time: 200 microseconds

**Ausgabe NMCU V3:**

AES-128-Random-Text

Average Encryption Time: 95 microseconds

Average Decryption Time: 142 microseconds

AES-192-Random-Text

Average Encryption Time: 114 microseconds

Average Decryption Time: 171 microseconds

AES-256-Random-Text

Average Encryption Time: 133 microseconds

Average Decryption Time: 200 microseconds

**Aufgabe 7**

Wie bei Aufgabe 6 wurde derselbe Code von Test\_AES benutzt und diesmal “perfRandomTextAndKeyCipher” Funktion hinzugefügt. Die Zeiten unterscheiden sich zwischen der Aufgabe 6 und 7 kaum voneinander.

**Ausgabe NMCU V2:**

ES-128-Random-Text-Key

Average Encryption Time: 96 microseconds

Average Decryption Time: 143 microseconds

AES-192-Random-Text-Key

Average Encryption Time: 114 microseconds

Average Decryption Time: 171 microseconds

AES-256-Random-Text-Key

Average Encryption Time: 133 microseconds

Average Decryption Time: 200 microseconds

**Ausgabe NMCU V3:**

ES-128-Random-Text-Key

Average Encryption Time: 96 microseconds

Average Decryption Time: 143 microseconds

AES-192-Random-Text-Key

Average Encryption Time: 114 microseconds

Average Decryption Time: 171 microseconds

AES-256-Random-Text-Key

Average Encryption Time: 133 microseconds

Average Decryption Time: 200 microseconds

**Aufgabe 8**

Für diese Aufgabe wurde aus demselben Repository (<https://github.com/OperatorFounda9on/Crypto/tree/master/examples>) der Test\_CTR Code, Test\_EAX, Test\_GCM, Test\_GCM undTest\_XTS benutzt. Es mussten lediglich passende Test Vektoren hinzugefügt und mit dem perfCipherEncrypt und perfCipherDecrypt Funktionen aufgerufen werden.

--------kommentar zum unterschied von v2 und v3 hinzufügen ------------

**CTR-Mode**

Testvektoren für 192 und 256 Bit hinzugefügt, indem wir die Keys passend erweitert haben.

**Ausgabe NMCU V2:**

**Performance Tests:**

**AES-128-CTR Encrypt ... Average Encryption Time: 815 microseconds**

**AES-128-CTR Decrypt ... Average Decryption Time: 816 microseconds**

**AES-192-CTR Encrypt ... Average Encryption Time: 815 microseconds**

**AES-192-CTR Decrypt ... Average Decryption Time: 816 microseconds**

**AES-256-CTR Encrypt ... Average Encryption Time: 815 microseconds**

**AES-256-CTR Decrypt ... Average Decryption Time: 816 microseconds**

**Ausgabe NMCU V3:**

Ausgabe v3  
-

-

-

-

-

**EAX-Mode**

**Ausgabe NMCU V2:**

**Ausgabe NMCU V3:**

**GCM-Mode**

In diesem Code sind schon alle nötigen Testvektoren vorhanden. Man muss nur den Code zum Laufen bringen, indem man die „crypto\_feed\_watchdog()“ auskommentiert. Zudem muss man alles, was mit importierten Libraries wie „Speck.h“, „SpeckTiny.h“ und „CryptoLW.h“ entfernen. Zum Schluss wurden die Print Ausgaben alle so angepasst, dass man die Durchschnittszeit bekommt.

**Ausgabe NMCU V2:**

Performance Tests:

AES-128 GCM #1 SetKey ... Average SetKey Time (100): 230 microseconds

AES-128 GCM #1 Encrypt ... Average Encryption Time (100): 1224 microseconds

AES-128 GCM #1 Decrypt ... Average Decryption Time (100): 1220 microseconds

AES-128 GCM #1 AddAuthData ... Average AddAuthData Time (100): 428 microseconds

AES-128 GCM #1 ComputeTag ... Average Compute Tag Time (100): 60 microseconds

AES-192 GCM #10 SetKey ... Average SetKey Time (100): 266 microseconds

AES-192 GCM #10 Encrypt ... Average Encryption Time (100): 1375 microseconds

AES-192 GCM #10 Decrypt ... Average Decryption Time (100): 1372 microseconds

AES-192 GCM #10 AddAuthData ... Average AddAuthData Time (100): 428 microseconds

AES-192 GCM #10 ComputeTag ... Average Compute Tag Time (100): 60 microseconds

AES-256 GCM #16 SetKey ... Average SetKey Time (100): 316 microseconds

AES-256 GCM #16 Encrypt ... Average Encryption Time (100): 1527 microseconds

AES-256 GCM #16 Decrypt ... Average Decryption Time (100): 1524 microseconds

AES-256 GCM #16 AddAuthData ... Average AddAuthData Time (100): 428 microseconds

AES-256 GCM #16 ComputeTag ... Average Compute Tag Time (100): 60 microseconds

**Ausgabe NMCU V3:**

**XTS-Mode**

**Ausgabe NMCU V2:**

**Ausgabe NMCU V3:**

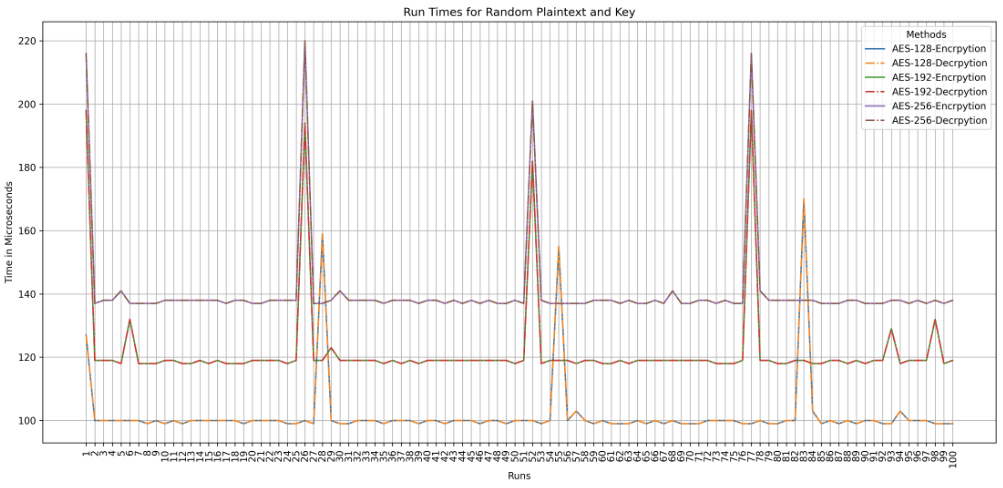
**Blatt 2**

**Aufgabe 2 und 3**

Die .csv Dateien sind mit in der Zip Datei enthalten.

Ein Bild, das Reihe, Text, Diagramm, parallel enthält.

Automatisch generierte Beschreibung**Random Plaintext**

**Random Plaintext and Key**

Ein Bild, das Text, Reihe, Diagramm enthält.

Automatisch generierte Beschreibung**AES CTR-Mode**