C coding exercise

Version 1.0

Our rationale

- As part of Escrypt's application process, we kindly ask you to solve a C programing exercise (max. 1 day effort)
  - You will obtain the exercise material approximately one week before the application interview
  - You can prepare your solution at home without stress
  - You shall give a short presentation of your results (e.g. source code walkthrough and maybe presentation of design) as part of the job interview
- The goal of this exercise is to stimulate a technical discussion between you and your potential future colleagues during the job interview (instead of asking the same standard questions again and again)

Your task

Background information

- A customer asks you to implement the so-called **AES encryption algorithm in counter mode**[1] (AES CTR) **for 128 bit keys**

- For this purpose, you obtain the customer's crypto library[1] (see `tiny-AES128-C-master.zip`)
  - This library already provides an implementation of **AES encryption in ECB mode** for 128 bit keys (function `AES128_ECB_encrypt`)
  - You can **use this function as a main building block for your own AES CTR implementation**

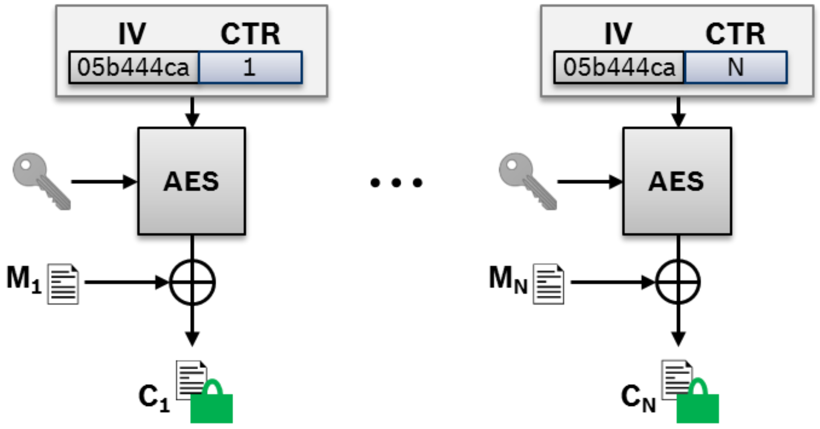
1 See attached document sp800-38a.pdf, taken from [NIST], for the official specification of AES CTR. Additional information is given in [Wiki]

2 This is a public domain library (see [AES_C] for more information). You may use the included test.c file to figure out how to invoke the AES_128_ECB_encrypt function. You may also modify / optimize the library if necessary.

Your task

How AES CTR works

- The AES CTR mode works as follows (for encryption):
  1. Compute a key stream by applying the AES algorithm to a internally maintained counter value (possibly combined with a random initialization vector/nonce as shown below)
  2. XOR the key stream with the plaintext message blocks[1]



1 It is up to you to decide how to handle input messages with message length that is *not* a multiple of the AES block length

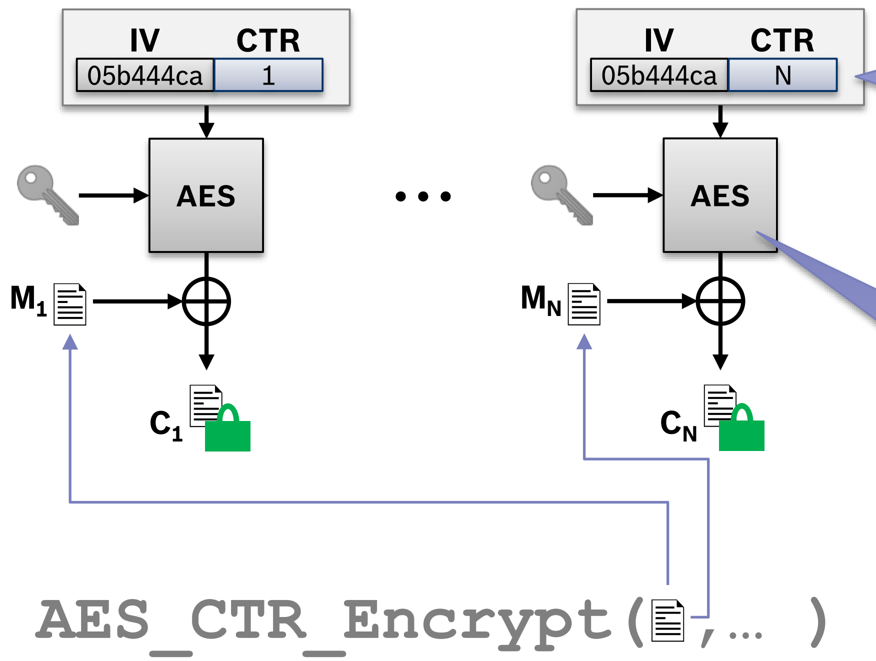
Your task
Further information

- You should implement at least the following APIs for AES CTR encryption and decryption with the following inputs[1]:
- Encryption:
  - A plaintext message (of arbitrary length[2])
  - The length of the plaintext message
  - A 128 bit key
- Decryption:
  - A ciphertext message (of arbitrary length[2])
  - The length of the ciphertext message
  - A 128 bit key

1 It is up to you to extend the API or define auxiliary functions, e.g. in order to allow encapsulation of the CTR mode's counter value, to include random initialization vectors or to handle the function's output value
2 It is up to you to define a maximal limit for supported input length if required by your implementation

Your task

Illustration



AES_CTR_Encrypt(, …)

You can design and implement your own appropriate counter handling.

You can use AES128_ECB_encrypt(…) here!

Your task
Hints & problem solving

- One way to implement a unique counter value is to concatenate a random nonce with a simple incremental counter value.
  Hint: In this case, the AES CTR decryption may require additional input.
- In case you have trouble with solving the exercise or in case you need additional information, please contact Alexander Meurer (alexander.meurer@escrypt.com)

Your task

Out of scope

- To keep the implementation effort low, the following topics are **out-of-scope** of this exercise:
  - You do *not* need to implement extensive testing for your implementation (however, doing some testing of the basic functionality is desirable)
  - You do *not* need to follow dedicated coding standards or guidelines such as MISRA-C or CERT secure coding (however, it is "nice-to-have" some basic rules considered)

References

- [NIST] – NIST special publications
  http://csrc.nist.gov/publications/PubsSPs.html
- [Wiki] Wikipedia – Block cipher mode of operation
  https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation
- [AES_C] - Tiny AES128 in C
  https://github.com/kokke/tiny-AES128-C

ESCRYPT - Embedded Security Headquarters

Lise-Meitner-Allee 4
44801 Bochum

Tel.: +49 234 43870-200
Fax: +49 234 43870-211

info@escrypt.com
www.escrypt.com