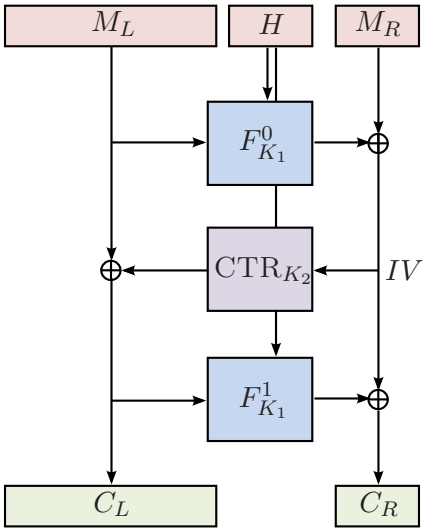
Problem Set 7
Course **Secure Channels**
(Summer Term 2016)

Bauhaus-Universität Weimar, Chair of Media Security
Prof. Dr. Stefan Lucks, Eik List
URL: `http://www.uni-weimar.de/de/medien/professuren/mediensicherheit/teaching/`
Due Date: 12 July 2016, 1:30 PM, via email to `eik.list(at)uni-weimar.de`.

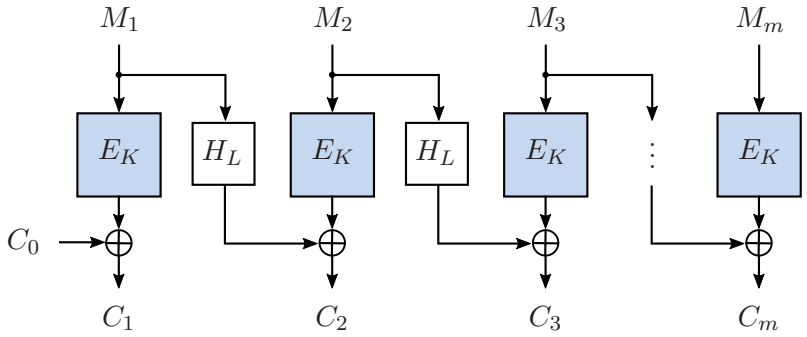
Task 1 – RIV (4 Credits)
Let $F : \{0,1\}^k \times \{0,1\} \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^n$ be a secure PRF, and $\mathrm{CTR} : \{0,1\}^k \times \{0,1\}^n \times \{0,1\}^* \to \{0,1\}^*$ be the CTR mode with some secure block cipher internally. Let $K_1, K_2 \in \{0,1\}^k$ denote two independent secret keys.
Alice tried to make RIV a little more efficient and proposes a variant RIV′. Given message M and header H, RIV′ encrypts and authenticates as follows:

- $M' \leftarrow$ Prepend n zero bits to M.
- Split M' into $M_L \,\|\, M_R$, where M_R consists of the last n bits of M and M_L consists of the rest of M.
- Use M_R as input to the right side and encrypt as RIV.
- The result $C_L \,\|\, C_R$ builds the authenticated ciphertext, i.e. including the tag.

Show *either* why the modified RIV′ construction is also secure under release of unverified plaintext *or* show an efficient attack on privacy or authenticity.



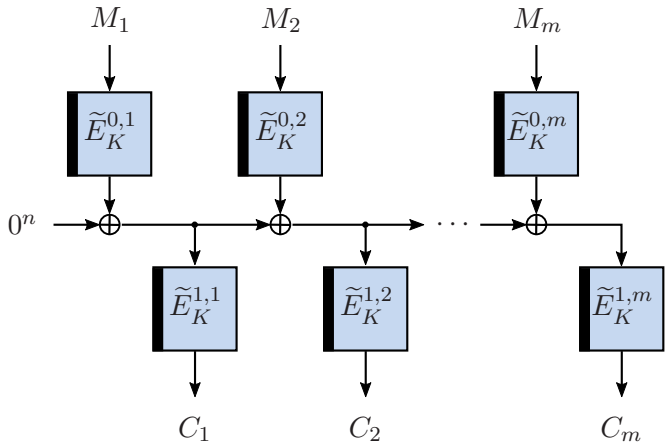
Task 2 – HCBC1′ (4 Credits)
The on-line cipher below is a mirrored version of HCBC1, which we call HCBC1′, hereafter. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ denote a secure block cipher, $H : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ an ϵ-AXU hash function, and let $K, L \in \{0,1\}^k$ denote two random and independent secret keys. Describe *either* an attack on HCBC1′ in the RoR-OPRP-CPA model *or* explain why it is secure.



Task 3 – Simple COPE (5 Credits)
Consider the following simplified version of COPE shown below. We only consider messages whose length is a multiple of the block length. Instead of the original COPE, we use a tweakable block cipher $\widetilde{E} : \{0,1\}^k \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$, with tweak set $\mathcal{T} := \{0,1\} \times \{0,1\}^{n-1}$ and secret key $K \in \{0,1\}^k$. So, our simplified version replaces all masks with distinct tweaks.

a) Explain if our simplification affects the security of the construction. Does it make COPE more, equally, or less secure?

b) Describe *either* an attack in the RoR-OPRP-CCA model, where one can ask plaintext and ciphertext queries, *or* explain why this version of COPE is secure.



Task 4 – Implementation (5 Credits)
Implement a simplified version of McOE-X [1] with AES-128 in Python. Use AES-128 as block cipher, e.g. from the `pycrypto` module. Implement tag splitting for messages whose length is not a multiple of 128 bits. Your program should use an API similar to that below. It does not have to be particularly efficient, but should as readable as possible.

```
#!/usr/bin/env python
from crypto import cipher

class McOEX:

  def __init__(self):
    # TODO: Create the cipher

  def setup(self, k):
    """Initializes the cipher's keys."""
    # TODO

  def encrypt_and_authenticate(self, n, a, m):
  """Encrypts the message and returns concatenated, ciphertext and tag
     (C || T), stores the authentication tag T that corresponds to the given
     tuple of nonce, associated data, and message (N, A, M)."""
    # TODO

  def decrypt_and_verify(self, n, a, c):
    """Given a tuple of nonce, associated data, ciphertext (N, A, C),
       decrypts it and returns the corresponding plaintext M if and only if
       C is authentic. Returns None and no information about M if any
       error occurs."""
    # TODO
```

References

[1] Ewan Fleischmann and Christian Forler and Stefan Lucks and Jakob Wenzel: McOE: A Family of Almost Foolproof On-Line Authenticated Encryption Schemes. Full Version of the appeared paper appeared in FSE 2012. Accessible from `https://eprint.iacr.org/2011/644.pdf`.