Technische Universität München
Lehrstuhl Informatik VIII
Prof. Dr.-Ing. Georg Carle
Dr. Heiko Niedermayer
Cornelius Diekmann, M.Sc.
Dr. Ralph Holz

Network Security Winter Term 2014/2015
Exercise 2

Exercise 1 Understanding crypto primitives

Assume a client-server architecture with an attacker on the data path. We define an attacker model in which the attacker is able to do the following:

- Eavesdrop on messages, i.e. listen to the transmission as a sequence of bits
- Modify messages
- Delay messages
- Delete messages

We assume a protocol with mechanisms for the following security services:

- Confidentiality by using symmetric cryptography
- Message integrity, applied per message, using MACs
- Authenticity of the sender

These are the *only* mechanisms the protocol supports.

a) The designers of the protocol are very lazy and want to use the same symmetric key for encryption and the MACs. They are going to use AES-CBC for both. What weaknesses are they going to introduce?

b) Which of the attacks in our attacker model are reliably detectable by the receiver? Give reasons.

c) The designers also want to detect message replay, i.e. the attacker storing and later forwarding a message. How can that be added to the protocol?

Exercise 2 Hash functions and passwords

a) Many multi-user systems do not store user passwords as plain text. Instead, they compute $p' = \mathrm{SHA1}(r|p)$, i.e. a SHA1 hash of a random string r to which the password is concatenated. They then store (r, p'). What is achieved with this?

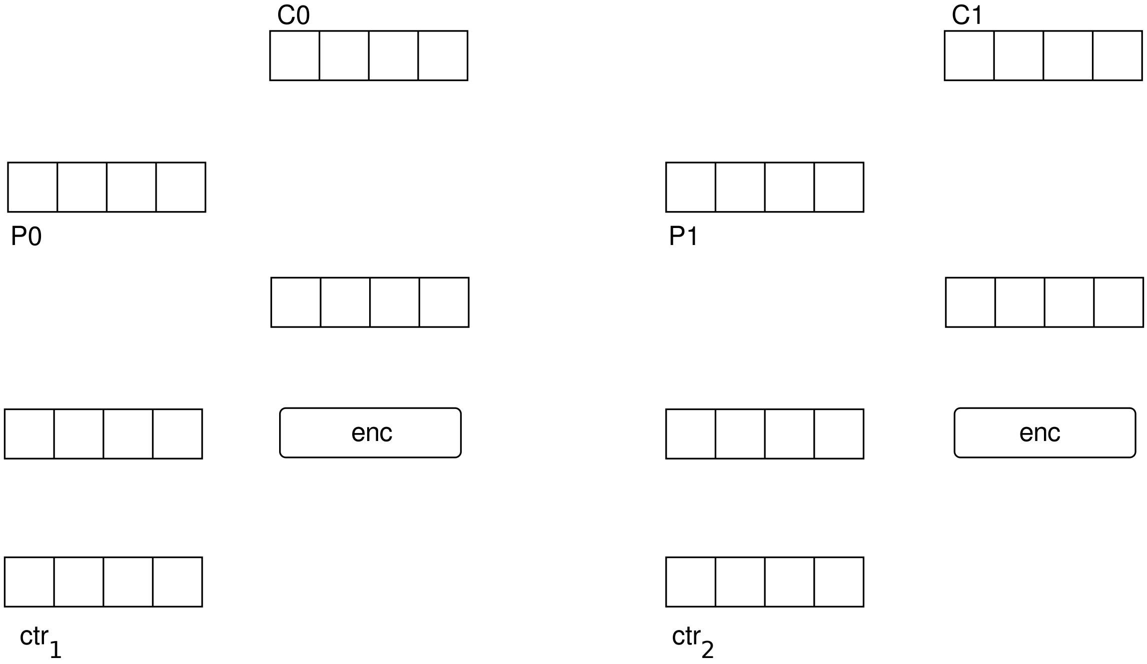
b) Why is it not a problem if r is known to an attacker?

c) Assume you own a company *Business Doping Inc.*, or *B-Dope* for short. Two engineers turn up for a job interview. You ask both the same question: 'How should I store user passwords?' Engineer A says: 'Store the passwords as hash values (as described in a)).' B says: 'Choose a global key of length 256 bits and use AES to encrypt the passwords individually, using ECB mode.' Following sound cryptographic considerations, which engineer should you hire – and why? Give 2 (cryptographic) reasons.

Exercise 3 Symmetric cryptography and hash functions

The following subproblems are all about symmetric crypto and hash functions. (Exam 2013)

a) For the following question, you will need Algorithm 1 in conjunction with Table 1. Connect the boxes below and add elements where necessary to sketch the operation of **Counter Mode (CTR)**. Then, apply CTR using `simpleCipher` as block cipher with 4 bits as block length on the plaintext $P = 10101100$ with key $k = 1011$. Let the nonce be 00. The running counter i has two bits and starts at 1. The counter (ctr_i in the figure) used for CTR Mode is constructed as the concatenation of nonce and i. The resulting ciphertext blocks are C0 and C1.



```
Algorithm 1 simpleCipher(key, m)
Require: key and m are 4 bits long
  z = key XOR m
  return lookuptable(z)
```

```
Algorithm 2 simpleHash(p)
Require: p of arbitrary length                              //Example $p = 00010100$
  $a$ = binary_to_decimal($p$)                              //Example $a = 20$
  $b = (a \cdot 3) \mod 16$                                 //Example $b = 12$
  $c$ = decimal_to_binary($b$)                              //Example $c = 00001100$
  return extract_the_four_least_significant_bits($c$)       //Example result: 1100
```

Input	Output
0000	0011
0001	0100
0010	0101
0011	0110
0100	0111
0101	1001
0110	1101
0111	1111
1000	1100
1001	1110
1010	1011
1011	1010
1100	1000
1101	0001
1110	0010
1111	0000

Table 1: lookupTable(z) for `simpleCipher`.

b) In the following, you will need Algorithm 2. Disregarding that its input and output length is too small, is `simpleHash` a cryptographic hash function? Give a proof or counterexample.

c) In the following, you will need Algorithm 2 again. You are given the message $P = 1010$, the hash function $h =$ `simpleHash` and the values $opad = 1001$, $ipad = 1100$ and a key $k = 1111$. Give the formula and compute the HMAC of P using hash function h.

d) Let h be a cryptographically secure hash function. Explain how you use it to encrypt text in a stream cipher-like mode.