Problem Set 1
Introduction to Modern Cryptography
Coursera-Course (Winter 2017)

Bauhaus-Universität Weimar, Chair of Media Security
Problem Session: Eik List, Course: Prof Dan Boneh, Stanford University.
URL: `http://www.uni-weimar.de/de/medien/professuren/mediensicherheit/teaching/`

Due Date: Friday, 24 Nov 2017, 1:30 PM, via email to `eik.list(at)uni-weimar.de`.

Question 1 – Stream Cipher (4 Points)
The stream cipher can be generalized to stream cipher based on alphabets rather than the binary one.

- Define a stream cipher which operates on the letter, $A, B, \cdots Z$, and is represented by the numbers $0, 1, \ldots, 25$. What is the encryption and decryption algorithms, and the key stream?
- Decrypt the following ciphertext: “`bsaspp mmuosp`” with “`rsidpy fmawoy`” as a key.

Question 2 – One-Time Pad (2 Points)
Assume an OTP-like encryption which uses a short key of 80 bit. We use this key to encrypt the large volume of data. Is this scheme secure? Explain your attack which breaks the scheme.

Question 3 – PRGs (4 Points)
Let $G : \{0,1\}^k \to \{0,1\}^n$ be a secure pseudo-random bitstream generator (PRG) that takes a k-bit seed K and outputs an n-bit pseudorandom value. Which of the following functions G' are also secure PRGs? Briefly explain your answer.

- $G'(K) := G(K) \vee 1^n$ (1^n denotes an n-bit string 11111..) and $\vee$ the bitwise OR.
- $G'(K) := G(0^k)$.
- $G'(K) := G(K) \wedge G(K)$, where $\wedge$ denotes bitwise AND.
- $G'(K, K') := G(K) \wedge G(K')$, where K and K' are independent uniformly at random chosen k-bit seeds.

Question 4 – One-Time Pad (4 Points)
Oscar has two ciphertexts (C_1 and C_2) which are encrypted by the same key. He also has XOR difference of two ciphrtext as a decimal number.

$C1 \oplus C2 =$
```
[17, 4, 1, 26, 0, 24, 23, 23, 10, 1, 28, 19, 19, 15, 20, 30, 6, 11, 4, 8, 17, 29,
23, 1, 24, 23, 28, 17, 20, 18, 9, 27, 17, 13, 13, 10, 13, 18, 7, 4, 22, 23, 10,
22, 13, 27, 8, 5, 28, 1, 23, 26, 19, 12, 6, 0, 8, 7, 2, 15, 0, 3, 7, 0, 9, 7, 29,
19, 19, 3, 2, 29, 27, 8, 11, 7, 0, 6, 17, 26, 10, 26, 31, 26]
```

Reconstruct the two plaintexts P_1 and P_2. To calculate the XOR difference of the alphabet use 5 bits, i.e $A = \texttt{00001}, B = \texttt{00010}, C = \texttt{00011}, \ldots, Z = \texttt{11010}$.

Tip: Oscar knows that the word `Kryptographie` is one of the plaintext word. He also knows that the string `Sicherheit` appears two times in the other plaintext.

Note: You can write a Python program to solve this task or you can do it by hand. You need to send me all the details for any method that you use.

Question 5 – 4DES Encryption (4 Points)

Since the DES uses only 56-bit keys, Alice wants to construct a cipher with higher security. She defines $4DES : \{0,1\}^{56} \times \{0,1\}^{56} \times \{0,1\}^n \to \{0,1\}^n$, which uses a 112-bit key, consisting of two independent secret 56-bit keys K_1 and K_2, and four calls of the original DES:

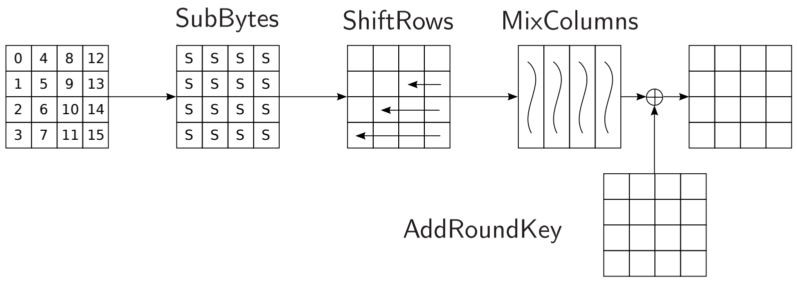
$$4DES_{K_1,K_2}(M) = DES_{K_2}(DES_{K_2}(DES_{K_1}(DES_{K_1}(M)))).$$

a) Describe a key-recovery attack on $4DES$ that takes significantly less time than testing all 2^{112} keys. Specify your attack time and memory complexities.

b) Describe (briefly) how one could modify 4DES such that the attack from a) are made infeasible.

Question 6 – AES (4 Points)

The AES is the standard block cipher nowadays: each of its 10 rounds consists of SubBytes, ShiftRows, MixColumns and AddRoundKey (except for the last round). Below you find an illustration of one round. The input on the left also shows the order of the bytes in the state.



This task shows what would happen if an operation missed. Let $P, P' \in \{0,1\}^{128}$ be two plaintexts that differ only in their byte 0 (here encoded in hexadecimal):

$$P = \texttt{0x0001020304050607 08090a0b0c0d0e0f},$$
$$P' = \texttt{0x}\underline{\texttt{ff}}\texttt{01020304050607 08090a0b0c0d0e0f}.$$

In the following, both plaintexts were encrypted with three different versions of the AES:

1. An AES version where MixColumns was omitted in every round.

2. An AES version where ShiftRows was omitted in every round.
3. A complete AES version.

Below you find three ciphertext pairs, encoded in hexadecimal. For all pairs, C_i is the ciphertext of P and C_i' that of P'. Find out which ciphertext pair (C_i, C_i') was generated by which AES version from above and explain your answer briefly.

$C_1 =$ `0x231446982181c5ca6476f957632cbd2a`, $C_1' =$ `0xbbac089c1cc8ff7e83f147177b488316`

$C_2 =$ `0xceee58e9437269400fc7e2466b25564c`, $C_2' =$ `0xb7ee58e9437269400fc7e2466b25564c`

$C_3 =$ `0x2b45b04bce63e0c1e750ca0c876248a9`, $C_3' =$ `0xb6012b6ece63e0c1e750ca0c876248a9`

Question 7 – Block-Cipher Modes (4 Points)
Alice wants to send the two-block (32 characters) message

$M = (M_1, M_2) =$ `Send␣to␣Bob␣100,-␣EUR␣from␣ Alice`

with $M_1, M_2 \in \{0,1\}^n$ encoded as 8-bit ASCII string encrypted to her bank. Alice chooses an initial value $IV \in \{0,1\}^n$, encrypts M with AES-128 in some mode and her secret key, and transmits the resulting ciphertext (IV, C_1, C_2) to her bank.
An adversary Eve intercepts Alice's ciphertext. Instead of the original text, Eve wants to replace it with a manipulated ciphertext $C' = (IV', C_1', C_2')$ for the message

$M' = (M_1', M_2') =$ `Send␣to␣Eve␣500,-␣EUR␣from␣Alice`

a) Specify a possible ciphertext (IV', C_1', C_2') for M' when the used mode is Counter mode.

b) Specify a possible ciphertext (IV', C_1', C_2') for M' when the used mode is CBC.

Hint: Note that Eve can freely choose a new initial value IV'.

Question 8 – Simple MACs (1+1+2 Points)
For each of the following MACs, briefly explain why they are secure or show how to efficiently forge the authentication code for a message. Prior, you can ask for the authentication of up to three chosen messages.

a) $MAC_K(M) := \bigoplus_{i=1}^{m} E_K(M_i)$.

b) $MAC_K(M) := \bigoplus_{i=1}^{m} E_K(M_i \oplus \langle i \rangle)$, where $\langle i \rangle$ denotes the n-bit representation of i.

c) $MAC_K(M) := \bigoplus_{i=1}^{m} E_{K \oplus \langle i \rangle}(M_i)$.

Question 9 – CBC-MAC' (4 Points)
Assume that $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ is a secure block cipher and $K_1 \in \{0,1\}^k$ a secret key. Let $K_2 \in \{0,1\}^n$ be a second independent key. We consider the following variant of CBC-MAC, called CBC-MAC'. Given an m-block message $M = (M_1, \ldots, M_m)$, CBC-MAC' computes an authentication tag as follows:

$$\begin{aligned} C_0 &= 0^n, \\ C_i &= E_{K_1}(C_{i-1} \oplus M_i), \qquad \text{for } 1 \leq i \leq m, \\ \text{CBC-MAC}'_{K_1,K_2}(M) &:= C_m \oplus K_2, \end{aligned}$$

Show how to efficiently predict the tag for a message with CBC-MAC'. You may ask for the tags of at most three (other) messages before. Note that you can vary the lengths of your chosen messages.