CS 519
Cryptography and Network Security
Instructor: Ali Aydın Selçuk
Department of Computer Engineering, Bilkent University

Homework #2

Due October 11, 2012, beginning of the class

1. Study the Rijndael (AES) encryption function and the design decisions taken to optimize it in implementation from the handout on the class webpage.

    (a) Explore the general structure of Rijndael. Is it a Feistel cipher? An SP cipher? Why/why not? Explain briefly.

    (b) Explain the functionality of each of the four basic operations, ByteSub, ShiftRow, MixColumn, AddRoundKey in the cipher.

    c) See how the round function can be implemented by just four table lookups and four XORs on a 32-bit platform. Discuss which properties of the diffusion part make this feature possible. (For instance, would a similar feature be possible if a bit-wise permutation were used for diffusion? Or, if a non-linear operation with no matrix representation were used for MixColumn? Or, what if a "MixRow" similar to MixColumn were used instead of ShiftRow?) Is the same trick applicable on an 8- or 16-bit platform? Why/why not?

    d) How is it achieved to have the decryption operation have the same structure as encryption? (That is, what features in the design of the round function, or the different operations at the initial and final rounds, etc. make this possible?)

2. Question 3, the midterm exam of Fall 2011.

3. Consider the following MAC scheme with a 128-bit block cipher, similar to the CTR mode of encryption:

    - Append a 1 bit to the message, and then enough 0 bits to make the total length of the message a multiple of 128 bits.

    - Break the message into 128-bit blocks; $M_1, M_2, \ldots$.

- Encrypt independently each block M_i as $C_i = E_K(M_i) \oplus E_K(i)$.
- Take the low-order two bytes of each ciphertext block C_i. Concatenate them, yielding the final MAC.

(Note that this is a variable-size MAC.)

a) Why is the padding done the way it is rather than just padding the last block with 0 bits?

b) Show that this MAC is not secure by a simple truncation attack. (I.e.: The attacker observes a message M and its MAC $MAC_k(M)$. Then he obtains the MAC of another message M', which is a truncation of M.)

c) Describe another attack on this MAC scheme where the attacker computes the MAC of a message without having the MAC of any other messages beforehand.