Written Exam
Cryptology

Department of Mathematics and Computer Science
University of Southern Denmark

Monday, January 29, 2007, 9:00–13:00

You are allowed to use the textbook and any notes you have for this course, along with a pocket calculator.

The exam consists of 7 problems on 5 numbered pages (1–5). All parts of all seven questions should be answered. The weight assigned to each problem in grading is given in parentheses at the start of each problem.

You may refer to algorithms and results from the textbook or problems which have been assigned during the course. In particular, you may give as a reason for a claim holding that it follows from a result in the textbook (assuming this is true). References to other books than the textbook will not be accepted.

Note that if there is a question in a problem which you cannot answer, you may continue with the following questions, assuming the result from the question you could not answer.

Problem 1 (10%)

Suppose that a keystream S is produced by a linear feedback shift register with m stages (by a linear recurrence relation of degree m).

a. Assume that S does not end with an infinite string of consecutive ones. What is the longest string of consecutive ones which could appear in S? Prove your answer.

b. Prove that it is possible for such a keystream S produced by an m-stage linear feedback shift register to have a period of m.

Problem 2 (15%)

Consider the following proposal for a hash function, where $E(K, M)$ is encryption of the 128 bits of M using a 128-bit key K in Rijndael (AES). Let IV be a 128-bit random string. Pad the document to be hashed with zeros so that the number of bits is divisible by 128. Let the resulting document be $M = m_1||m_2||...||m_r$, where each block m_i contains exactly 128 bits and the operation $||$ is concatenation.

$$\begin{aligned} H_0 &\leftarrow IV \\ H_1 &\leftarrow E(m_1, H_0) \\ H_2 &\leftarrow E(m_2, H_1) \\ &\vdots \\ H_r &\leftarrow E(m_r, H_{r-1}) \\ H &\leftarrow E(m_1 \oplus m_2 \oplus ... \oplus m_r, H_r) \end{aligned}$$

The operation $\oplus$ is bit-wise exclusive-or, and the output of the hash function is (H_0, H). (Note that there a message which has zeros at the end will hash to the same value as that message with the zeros truncated (removed). Thus, finding collisions is trivial, but we will ignore that type of collision in the following.)

a. Using the Birthday Paradox, define and analyze (how many calls to Rijndael) an algorithm for finding a collision.

b. The above hash function would be much less secure if the steps with the ByteSub transformations were simply removed from Rijndael. Which is the hardest of the three problems Preimage, Second Preimage, and Collision, which could now be solved efficiently? How would you solve that problem?

Problem 3 (5%)

Suppose the El Gamal Public-key Cryptosystem in Z_p^* is used to encrypt many blocks of a message, and the different secret numbers k used (one for each block) are generated using a pseudo-random bit generator. (The value k used in encrypting the ith block is the ith block of 512 bits output by the pseudo-random bit generator.)

Why must the pseudo-random bit generator have a long period?

Problem 4 (10%)

a. Show all steps in the calculations of the Jacobi symbol $\left(\frac{29}{35}\right)$, using the standard algorithm (using the four properties of the Jacobi symbol given in the textbook).

b. Show all steps of the execution of one call to the Solovay-Strassen Primality Test, checking if 35 is prime. Assume that the random integer a chosen is 19.

Problem 5 (20%)

Suppose the Solovay-Strassen Primality Test is used to find the primes p and q for use in the RSA cryptosystem. (Assume that random integers of the required length are chosen and tested for primality until two are found where the test does not discover that they are composite.) Even assuming that the primality test is executed several times, there is still a small probability of choosing a number which is not prime. Suppose the p chosen is prime, but q is not.

a. Suppose q is a Carmichael number. (Recall that a *Carmichael number* is a composite integer n such that for all $x \in Z_n^*$, $x^{n-1} \equiv 1 \pmod{n}$.) Would encryption and decryption still work properly? Prove your answer. (Hint: try using the Chinese Remainder Theorem.)

b. Suppose q is a not a Carmichael number. Would encryption and decryption still work properly? Prove your answer. (Hint: try using the Chinese Remainder Theorem.)

c. What other problem could exist if q is a composite number?

Problem 6 (30%)

Let p be a large prime and let $x, y \in Z_p^*$. Suppose that $x = g^k \pmod{p}$ and $y = h^k \pmod{p}$. Assume the Prover knows the value k and that both the Prover and the Verifier are given the values p, g, h, x, and y. To show that the discrete logarithm of x with respect to g is equal to the discrete logarithm of y with respect to h, one can execute the following protocol:

Prover		Verifier
Choose random $s \in Z_{p-1}$. Let $u \equiv g^s \pmod{p}$ $v \equiv h^s \pmod{p}$.		
	$\xrightarrow{u, v}$	
		Choose a random $c \in Z_{p-1}$.
	$\xleftarrow{c}$	
Let $e \equiv c \cdot k + s \pmod{p-1}$.		
	$\xrightarrow{e}$	
		Check that $g^e \equiv x^c \cdot u \pmod{p}$ and $h^e \equiv y^c \cdot v \pmod{p}$. If so, accept. Otherwise, reject.

---

a. Prove completeness for the above protocol, showing that (assuming that both the Prover and Verifier follow the protocol) the Verifier will accept if the discrete logarithm of x with respect to g is equal to the discrete logarithm of y with respect to h.

b. Prove soundness for the above protocol. Assume that the discrete logarithm of x with respect to g is not equal to the discrete logarithm of y with respect to h. (Hint: after assuming that the Prover can give acceptable answers for two different values of c, show how a transcript containing both executions could be used to find the discrete logarithm of x with respect to g and the discrete logarithm of y with respect to h.)

c. Prove that the above protocol is honest verifier zero-knowledge, i.e., show that one can efficiently generate conversations $((u, v), c, e)$ with the same distribution as produced by the honest Prover and Verifier, without knowing k.

Problem 7 (10%)

Consider the following access structure:

$$\Gamma_0 = \{\{P_1, P_2\}, \{P_2, P_3, P_4\}, \{P_2, P_4, P_5\}, \{P_3, P_4, P_5\}\}$$

a. With this structure, should a coalition consisting of P_1, P_2 and P_4 be able to reconstruct the secret?

Should a coalition consisting of P_1, P_3 and P_4 be able to reconstruct the secret?

b. Explain how you would construct a secret sharing scheme for this access structure.