Examination in Modern Cryptology - 236506

Winter Semester, 2014-2015
3.2.2015, Moed A

Lecturer: Eli Biham
Teaching Assistant: Michal Dory

Exam Duration: Three hours.
The exam contains 4 questions. Answer all of them.

Only the formula sheet provided with the exam is allowed.
Any other study material is forbidden during the exam.

Devote the first 10 minutes for reading and understanding all the questions.

Make your answers as short and as clear as possible. Explain all your answers. Use as efficient algorithms as you can.

Allocate in advance two pages for each question, according to the order of the questions.

Write in an orderly and clean manner, with clear handwriting. Unclear answers will not be checked.

Good Luck!
Question 1 (25 points)

In this question we will show that if we could find the order of an element in $\mathbb{Z}_n^*$ efficiently, than we could find the factorization of $n$. We assume that $n = pq$ is the product of two large primes.

a. How many different square roots does 1 have in $\mathbb{Z}_n^*$? Show that finding a non-trivial square root of 1 allows to compute a non-trivial factor of $n$.

Let $a \in \mathbb{Z}_n^*$, and let $r$ be the order of $a$ in $\mathbb{Z}_n^*$.

b. Show that if $r$ is even, then $a^{r/2}$ is a square root of 1, which is different than 1.

Let $r_p$ be the order of $a \pmod{p}$ in $\mathbb{Z}_p^*$, and let $r_q$ be the order of $a \pmod{q}$ in $\mathbb{Z}_q^*$.

c. Show that $r_p | r$ and $r_q | r$.

Let:

\[
\begin{align*}
  r &= 2^k \cdot t \\
  r_p &= 2^{kp} \cdot t_p \\
  r_q &= 2^{kq} \cdot t_q 
\end{align*}
\]

Where $t, t_p, t_q$ are odd.

d. Deduce from c. that $k_p, k_q \leq k$, and show that $k = \max\{k_p, k_q\}$.

e. Use the Chinese Remainder Theorem and d. to show that if $k_p \neq k_q$ then $r$ is even and $a^{r/2} \equiv n \pm 1$.

It can be shown that for a random $a$, $k_p \neq k_q$ with high probability (greater than 0.5).

f. Assume that you have access to an oracle that returns the order of $a$ given an element $a \in \mathbb{Z}_n^*$. Describe an efficient probabilistic algorithm that finds the factorization of $n$, and describe it’s complexity.
Question 2 (25 points)

Let H be a hash function that its output is an element in a Group G of size m. I.e., $H: \{0,1\}^* \rightarrow G$.

a. What is the complexity needed in order to find a collision?

Assume also that for any two messages $M_1$, $M_2$ it holds that:

$$H(M_1 \cdot M_2) = H(M_1) \cdot H(M_2)$$

where $M_1 \cdot M_2$ is the product of $M_1$ and $M_2$ as integers, and $H(M_1) \cdot H(M_2)$ is the product of two elements in the group. Assume that it’s easy to find an inverse in the group.

b. Show how you can use this property and the birthday paradox in order to find a preimage. What is the complexity of the attack you described?

Assume now that $H: \{0,1\}^* \rightarrow \mathbb{Z}_p^*$, and that $H(M) = M^e \mod p$, for a prime $p$ and a fixed and known $e$, such that $\gcd(e, p - 1) = 1$.

c. Can you now find a collision with a smaller complexity than of a.? If your answer is yes, explain how. If your answer is no, explain why.

d. Can you now find a preimage with a smaller complexity than of b.? If your answer is yes, explain how. If your answer is no, explain why.

e. Explain how you can use the attack of b. in order to attack RSA. I.e., show how given a public key $(n, e)$ and a ciphertext $C = M^e \mod n$, you can find $M$ with a substantially smaller complexity than of an exhaustive search for $M$. What is the complexity of the attack you described?

f. Can you find an algorithm that finds the secret key with the same complexity as of e.? Explain.
Question 3 (25 points)

In this question we will explore the possibility of saving memory in a meet-in-the-middle attack on 4DES.

We will define 4DES as performing DES four times with four different keys, i.e.:

\[ C = DES_{K_4} \left( DES_{K_3} \left( DES_{K_2} \left( DES_{K_1} (P) \right) \right) \right) \]

a. What is the size of the key of 4DES?

b. Describe an exhaustive search algorithm for finding the key. What is its complexity? (Assume that you are given several pairs of blocks \( P, C \)).

c. Describe a meet-in-the-middle attack on 4DES. What are the time and memory complexities of the attack you described?

d. Assume that in addition to the pairs of blocks \( P, C \), you are given also the intermediate values after two DES encryptions (with the right keys). I.e., for each pair of blocks \( P, C \), you are also given the intermediate value \( X \), such that:

\[ X = DES_{K_2} \left( DES_{K_1} (P) \right) \]
\[ C = DES_{K_4} \left( DES_{K_3} (X) \right) \]

Describe an attack that uses this data, and is efficient as you can. What are the time and memory complexities of the attack you described?

e. Assume now that the intermediate value \( X \) is given only for one pair of blocks \( P, C \). Describe an attack that uses this data, and is efficient as you can. What are the time and memory complexities of the attack you described?

f. Assume now that the intermediate values are not given. Describe an attack where you guess all the options for the intermediate value \( X \) for the first block \( P, C \), and that for every guess you apply the attack from e. Does the attack work? What are the time and memory complexities? Do you save anything in comparison to a regular meet-in-the-middle attack? Explain.

g. A student from Stamford university claimed that according to the birthday paradox in order to find the key that encrypts \( P \) to \( C \), it’s enough to pick at random \( 2^{32} \) values for the pair of keys \( (K_1, K_2) \), and \( 2^{32} \) values for the pair of keys \( (K_3, K_4) \), and apply a meet-in-the-middle attack in order to find a collision in the middle that will give the key. What do you think about his suggestion?
**Question 4 (25 Points)**

Let $G = (V, E)$ be a graph. Hamiltonian cycle for a graph is a cycle in the graph that visits each vertex exactly once. Finding a Hamiltonian cycle for a graph is known to be a NP-complete problem.

Peggy wants to prove to Victor that she knows a Hamiltonian cycle for a given graph $G$, without revealing the cycle. So she went to wikipedia, where she found the following protocol:

**Protocol 1:**

Repeat $t$ times:
1. Peggy chooses a random permutation $\pi$ on $V$, and sends to Victor $G_1 = \pi(G) = (\pi(V), \pi(E))$.
2. Victor chooses a random $i \in \{1, 2\}$ and sends it to Peggy.
3. If $i = 1$, Peggy sends $\pi$ to Victor. If $i = 2$, Peggy reveals a Hamiltonian cycle in $G_1$.
4. Victor checks Peggy’s answer.

   a. Show that this protocol is an interactive proof for the Hamiltonian cycle problem. In particular, show how Peggy calculates her answer at 3., and how Victor checks her answer at 4.
   
   b. Show that this protocol is not Perfect Zero-Knowledge.

Victor suggested changing the protocol by using a commitment scheme:

**Protocol 2:**

Repeat $t$ times:
1. Peggy chooses a random permutation $\pi$ on $V$, and computes $G_1 = \pi(G) = (\pi(V), \pi(E))$.
2. For each of the $\frac{|V|(|V|-1)}{2}$ possible edges on $G_1$ Peggy computes a different commitment, and sends the commitments to Victor. I.e., for each pair of vertexes $(u_1, u_2) \in \pi(V)$, she sends commitment on 1 if $(u_1, u_2) \in \pi(E)$, and commitment on 0 otherwise.
3. Victor chooses a random $i \in \{1, 2\}$ and sends it to Peggy.
4. If $i = 1$, Peggy sends $\pi$ to Victor, and reveals all the commitments on the graph. If $i = 2$, Peggy reveals commitments on $|V|$ edges that form a Hamiltonian cycle in $G_1$.
5. Victor checks Peggy’s answer.

   c. What kind of commitment should we use (with perfect secrecy or perfect binding)? Explain.
   
   d. Is this protocol Zero-Knowledge? If your answer is yes, prove it, and determine if the protocol is Perfect Zero-Knowledge or Computational Zero-Knowledge. Otherwise, explain what Victor can learn from the protocol. You don’t have to show that the protocol is an interactive proof.