Advanced Cryptography — Midterm Exam

Serge Vaudenay
7.5.2012

- duration: 3h00
- any document is allowed
- a pocket calculator is allowed
- communication devices are not allowed
- the exam invigilators will not answer any technical question during the exam
- the answers to each exercise must be provided on separate sheets
- readability and style of writing will be part of the grade
- do not forget to put your name on every sheet!

1 Decryption Attack on Broadcast RC4

The RC4 pseudorandom number generator is defined by a state and an algorithm which update the state and produces an output byte. In RC4, a state is defined by

- two indices \(i\) and \(j\) in \(\mathbb{Z}_{256}\);
- one permutation \(S\) of \(\mathbb{Z}_{256}\).

By abuse of notation we write \(S(x)\) for an arbitrary integer \(x\) as for \(S(x \mod 256)\). The state update and output algorithm works as follows:

1: \(i \leftarrow i + 1\)
2: \(j \leftarrow j + S(i)\)
3: exchange the values at position \(S(i)\) and \(S(j)\) in table \(S\)
4: output \(z_2 = S(S(i) + S(j))\)

Q.1 Assume that the initial \(S\) is a random permutation with uniform distribution and that \(i\) and \(j\) are set to 0.

Q.1a What is the probability that \([S(1) \neq 2 \text{ and } S(2) = 0]\)?
Q.1b If \(S(1) \neq 2\) and \(S(2) = 0\) hold, show that the second output \(z_2\) is always 0.
Q.1c In other cases, we assume that \(z_2 = 0\) with probability close to \(\frac{1}{256}\).

Deduce \(p = \Pr[z_2 = 0]\). What do you think of this probability?

Q.2 Here, we let \(p = \Pr[z_2 = 0]\) and we assume that \(\Pr[z_2 = x] = \frac{1-p}{N-1}\) for all \(x \neq 0\) and \(N = 256\). We consider that a message \(m\) is encrypted by XORing to the stream generated by RC4. I.e., the ciphertext \(c\) is such that \(c_i = m_i \oplus z_i\). We assume that the same message \(m\) is encrypted \(n\) many times and that the adversary collected the ciphertext. Each encryption starts with an independent random permutation. Let \(n_x\) be the number of occurrences of the byte \(x\) in \(c_2\). I.e., there are \(n_x\) collected ciphertexts \(c\) such that \(c_2 = x\) in total.

Q.2a Compute the expected value of \(n_x\) for \(x = m_2\) then for any fixed \(x \neq m_2\).
Q.2b For \(x \neq m_2\) fixed, express \(n_{m_2} - n_x\) as a sum of \(n\) independent identically distributed (iid) random variables \(X_i\) which take values in \(\{-1, 0, 1\}\) and compute their expected value.
Q.2c We recall the Hoeffding bound:

**Theorem 1 (Hoeffding).** Let $X_1, \ldots, X_n$ be $n$ iid random variables which take values in $[a, b]$ and expected value $\mu$. For any $t > 0$, we have

$$\Pr \left[ \sum_{i=1}^{n} X_i \leq \mu - t \right] \leq e^{-\frac{2nt^2}{(b-a)^2}}$$

Give an upper bound for $\Pr[n_{m_2} \leq n_x]$ for any $x \neq m_2$.

Deduce an upper bound for the event that $n_{m_2}$ is not the largest counter value $n_x$.

Q.2d Propose an algorithm to decrypt $m_2$ and a lower bound on its probability of success.

What is the required number of ciphertexts to decrypt well almost certainly?

Propose a numerical application with the values from this exercise.

2 Generic Attacks on Multiple Encryption

We consider a block cipher $E$ with $n$-bit blocks and $n$-bit keys. We denote by $D$ the decryption algorithm. A $r$-time encryption is a process of encrypting a plaintext $P$ into $C = E_{k_1}(\cdots E_{k_r}(P)\cdots)$. We consider the problem of key recovery for a multiple encryption, with a few known plaintext/ciphertext pairs. I.e., we assume that the adversary knows some pairs $(P_i, C_i)$, for $i = 1, \ldots, r$, and want to find all $(k_1, \ldots, k_r)$ which would encrypt each $P_i$ to $C_i$.

In what follows, we consider the worst case complexity.

Q.1 Give an algorithm for $r = 1$. What are its time complexity and memory complexity?

Q.2 Give an algorithm for $r = 2$. What are its time complexity and memory complexity?

Q.3 We now consider $r = 4$.

Q.3a Given $P_1, P_2, B_1 \in \{0, 1\}^n$, how many $(B_2, k_1, k_2)$ triplets are such that $E_{k_2}(E_{k_1}(P_1)) = B_1$ for $i = 1, 2$?

Propose an algorithm with time-complexity $O(2^n)$ and memory complexity $O(2^n)$ to list them all.

Q.3b Given $P_1, P_2, B_1, C_1, C_2$ and a list of $(B_2, k_1, k_2)$ such that $E_{k_2}(E_{k_1}(P_1)) = B_1$ for $i = 1, 2$ from the previous algorithm, propose an algorithm to list all $(k_1, \ldots, k_4)$ such that $E_{k_4}(\cdots E_{k_1}(P_1)\cdots) = C_4$ for $i = 1, 2$ and $E_{k_2}(E_{k_1}(P_1)) = B_1$.

Q.3c Propose an algorithm with time-complexity $O(2^{2n})$ and memory complexity $O(2^n)$ to solve the key recovery problem.

Q.4 We now consider $r = 7$.

Q.4a Given $P_1, P_2, B_1, B_2 \in \{0, 1\}^n$, how many $(k_1, k_2, k_3)$ triplets are expected to satisfy the relations $E_{k_3}(E_{k_2}(E_{k_1}(P_1))) = B_i$ for $i = 1, 2$?

Propose an algorithm with time-complexity $O(2^{2n})$ and memory complexity $O(2^n)$ to list them all.

Q.4b Given $P_1, \ldots, P_7, B_1, B_2 \in \{0, 1\}^n$, propose an algorithm with time-complexity $O(2^{2n})$ and memory complexity $O(2^n)$ to list all $(B_3, \ldots, B_7)$ such that there exists a $(k_1, k_2, k_3)$ triplets such that $E_{k_3}(E_{k_2}(E_{k_1}(P_1))) = B_i$ for $i = 1, \ldots, 7$.

Q.4c By combining the algorithms of Q.4b and Q.3, propose an algorithm to do the key recovery for 7-multiple encryption, with time complexity $O(2^{4n})$ and memory complexity $O(2^n)$. 


3 Another Attack on Broadcast RSA

Q.1 Let \( N_1 = 235, N_2 = 451, N_3 = 391 \) be three RSA moduli, all working with the public exponent \( e = 3 \). Let \( y_1 = 99, y_2 = 238, y_3 = 278 \) be the respective encryption of the same \( x \) under the three RSA keys. Compute \( x \) without factoring any moduli.

Hint: \((N_2N_3)^{-1} \mod N_1 = 31, (N_1N_3)^{-1} \mod N_2 = 72, (N_1N_2)^{-1} \mod N_3 = 277\).

Q.2 Let \((N_i, e_i), i = 1, \ldots, r\) be \( r \) different RSA public keys, with pairwise coprime moduli. Let \( y_i = x^{e_i} \mod N_i \), for some positive \( x \) which is lower than all moduli. Let \( e = \max_i e_i \) and \( N = N_1 \cdots N_r \). We assume that an adversary knows all public keys and all \( y_i \) but not \( x \).

Q.2a Show that for each \( i \), there is a monic polynomial \( P_i(z) \) of degree \( e \) which can be computed by the adversary and such that \( P_i(x) \equiv 0 \pmod{N_i} \).

Q.2b Deduce that there is a monic polynomial \( P(z) \) of degree \( e \) which can be computed by the adversary and such that \( P(x) \equiv 0 \pmod{N} \).

Q.2c Deduce an algorithm to solve \( x \), for \( r \) large enough. How large?

We recall the Coppersmith result: Let \( f(z) \) be a monic polynomial of degree \( e \) in one variable modulo \( N \). There is an efficient algorithm to find all roots \( x \) such that \( 0 \leq x \leq N^{\frac{1}{e}} \).