Introduction to cryptology
TD#3

2024–W10

Exercise 1: Bad authenticated encryption

We consider a symmetric encryption scheme Enc and a *deterministic* MAC M (that always maps a given (key,message) pair (k, m) to the same tag t).

Q.1:

1. Show that $\mathsf{Enc} + \mathsf{M} : (k', k, m) \mapsto \mathsf{Enc}(k', m) || \mathsf{M}(k, m)$ has weak security w.r.t. the IND-CPA definition, *regardless of the IND-CPA security of* Enc (and under very mild assumptions on the UP security of M).

2. Propose an alternative way of combining Enc with a MAC in order to get an "authenticated" encryption scheme, and informally justify its IND-CPA security and resistance to forgeries.

Exercise 2: MAC security definitions *(Adapted from final exam '20)*

We again consider a deterministic MAC M.

Q.1: Assume that you know an algorithm $A^{\mathsf{U}}_{\mathsf{M}}$ that lets you win the universal forgery game for M with probability $p^{\mathsf{U}}_{\mathsf{M}}$, and let $t^{\mathsf{U}}_{\mathsf{M}}$ and $q^{\mathsf{U}}_{\mathsf{M}}$ respectively denote its running time and the number of queries it makes to its oracle.

1. Give a (possibly randomised) algorithm $A^{\mathsf{E}}_{\mathsf{M}}$ computing existential forgeries for M and that uses $A^{\mathsf{U}}_{\mathsf{M}}$ as a black box.

2. Give the cost $t^{\mathsf{E}}_{\mathsf{M}}$ and $q^{\mathsf{E}}_{\mathsf{M}}$ of your algorithm $A^{\mathsf{E}}_{\mathsf{M}}$, and its success probability $p^{\mathsf{E}}_{\mathsf{M}}$.

Q.2: We now assume the existence of $A^{\mathsf{E}}_{\mathsf{M}}$ as above.

1. Give a PRF adversary for M that uses $A^{\mathsf{E}}_{\mathsf{M}}$ as a black box, runs in time $t^{\mathsf{F}}_{\mathsf{M}} \approx t^{\mathsf{E}}_{\mathsf{M}}$ and makes $q^{\mathsf{F}}_{\mathsf{M}} \approx q^{\mathsf{E}}_{\mathsf{M}}$ queries to its oracle.

2. Deduce from that a lower-bound for $\mathbf{Adv}^{\mathrm{PRF}}_{\mathsf{M}}(q^{\mathsf{F}}, t^{\mathsf{F}})$.

3. Is the following (informally stated) scenario possible: "M is vulnerable to an existential forgery attack, yet is hard to distinguish from a random function"?

Q.3: We say that an assumption A_1 is *stronger* than an assumption A_2 if breaking A_2 implies breaking A_1 with a similar cost, but breaking A_1 does not necessarily imply breaking A_2 with a similar cost. Consider the three following (informally stated) assumptions: A_1: M is hard to distinguish from a random function; A_2: there is no efficient universal forgery attack on M; A_3: there is no efficient existential forgery attack on M.

1. Order the assumptions A_1, A_2, A_3 from weakest to strongest. *Be careful to justify your answer.*

2. Suppose that you need a MAC algorithm, and are magically given access to one that satisfies an assumption that you are free to choose; which of A_1, A_2 or A_3 would you pick (and why)?

RC4 is a stream cipher that can be used to (poorly) encrypt binary strings of arbitrary length in the following way:

1. Two communicating parties share a secret key k.
2. For each new plaintext p to be encrypted, one picks a unique initialisation vector v.
3. One runs a setup algorithm on the pair (k, v) that returns an initial state s (that depends on both k and v).
4. One runs the RC4 keystream generator on s, producing a keystream z of the same length as p.
5. The encryption of p is returned as $c := p \oplus z$, along with the initialisation vector v.

A designer suggests to use RC4 as the basis of a MAC algorithm. For simplicity, we assume that the input is at least 128-bit long, or that it has otherwise been padded up to that length (or longer) using an appropriate injective padding scheme. To authenticate a message one runs RC4 encryption on the input and returns the last 128 bits of the ciphertext as a tag. In more details:

1. Two communicating parties share a secret key k.
2. One runs a setup algorithm on the pair $(k, 0)$ that returns an initial state s.
3. For each new input x to be authenticated, one runs the RC4 keystream generator on s, producing a keystream z of the same length as x.
4. One encrypts x as $c := x \oplus z$; the last 128 bits of c are returned as the authentication tag of x.

Q.4:

1. Give (and analyse) a very efficient attack on RC4-MAC with respect to either one of the three security notions studied in this exercise.

Exercise 3: `tls-not-unique` *(Adapted from final exam '21)*

A certain network protocol authenticates every packet of 384 bits using a MAC that has tags of bitlength 96. For every *session* of the protocol (what is a session is not important here, but in a typical day one expects much more than 2^{40} sessions to be created worldwide), an identifier that is expected to uniquely identify the session among all possible sessions (past and future) is taken to be the 96-bit tag of a designated packet that is part of the session.

1. Identify a problem in the above process.
2. Propose a simple solution to fix it.

Exercise 4: Birthdays and a random sequence *(Adapted from final exam '17)*

Let $\mathcal{S}$ be a set of size N; let $(u_n)_{n \in \mathbb{N}}$ be a sequence whose elements are drawn independently and uniformly at random from $\mathcal{S}$, i.e. for all i, $u_i \twoheadleftarrow \mathcal{S}$. Suppose that you do not initially know $\mathcal{S}$,[1] nor N.

1. Give an algorithm that examines $\Theta(\sqrt{N})$ terms of (u_n) and that returns an approximation of N (you do not need to quantify the quality of this approximation).
2. What is the time and memory cost of your algorithm (be careful to fully specify the data structures you may use)?

[1] Be careful that the elements of $\mathcal{S}$ need not be integers. For instance $\mathcal{S}$ could be equal to $\{$*martes martes*, *martes foina*, *martes zibellina*$\}$.