Information Security class
Laboratory session 3

instructors: Nicola Laurenti, Laura Crosara

2023-24

Naïve entity authentication scheme

Your aim is to implement and evaluate the weakness of the following naïve challenge-response scheme for entity authentication with asymmetric cryptography

entities the prover A, the verifier B

setup Let p be a prime and α a primitive element in $\mathbb{Z}_p$, both publicly known. In the following $k \in \mathbb{Z}_p$ will denote the private key of A, while $k' \in \mathbb{Z}_p$ denotes the public key of A obtained from k as $k' = \alpha^k \mod p$. Assume that B knows k'.

$\boxed{1}$

A $\rightarrow$ B : $m = \mathrm{id}_\mathrm{A}$

$\boxed{2}$

B : generates a random and uniform challenge $c \in \mathbb{Z}_p$

B $\rightarrow$ A : c

$\boxed{3}$

A : generates $r' \sim \mathcal{U}(\mathbb{Z}_p)$ and computes the sum of its decimal digits, call the sum r. If $\gcd(r, p-1) \neq 1$ change r' and repeat until $\gcd(r, p-1) = 1$

A : computes $t_1 = \alpha^r \mod p$ and $t_2 = (c - kt_1)r^{-1} \mod (p-1)$

A $\rightarrow$ B : $t = (t_1, t_2)$

$\boxed{4}$

B : computes $s = \alpha^c \mod p$

B : computes $\hat{s} = k'^{t_1} t_1^{t_2} \mod p$, if $s = \hat{s}$ then A is accepted, otherwise A is rejected

Your tasks

1. Implement the protocol in a programming language of your choice. Evaluate its running time for several values of p between 10^3 and 10^7 (averaged over random choices of α and c).
2. An attacker C can observe some legitimate rounds of the protocol. In the file dataXxxxx.txt, where Xxxxx is your team's name, you can find pairs of eavesdropped messages c and t, all obtained with the same private key k. Design and implement an attack to the above protocol that allows C to successfully masquerade A.
3. Design and implement an attack that allows an attacker C to masquerade as A, observing only one previous run of the protocol. Evaluate through simulations the success probability of a single masquerade attempt. Then, assume that C is allowed to make n consecutive attempts (yet still having observed only one legitimate run between A and B), evaluate the probability of having one successful attempt for different values of n.

4. Now, change step 3 as follows:

   A : generates a random nonce $n \in \mathbb{Z}_p$ and computes $u = c + n \mod p$;
   A : generates $r' \sim \mathcal{U}(\mathbb{Z}_p)$ and computes the sum of its decimal digits, call the sum r. If $\gcd(r, p-1) \neq 1$ change r' and repeat until $\gcd(r, p-1) = 1$
   A : computes $t_1 = \alpha^r \mod p$ and $t_2 = (u - kt_1)r^{-1} \mod (p-1)$
   A → B : $t = (n, t_1, t_2)$

   and change step 4 accordingly. Design and implement an attack which allows C to successfully masquerade as A without observing any previous run and knowing only k'.

What you need to turn in

Each team must turn in, through the Moodle assignment submission procedure:

1. the source code for your implementation (either as a single file, many separate files, or a compressed folder)
2. a short report (to be submitted as a separate file from the source code file / compressed folder) in a graphics format (PDF, DJVU or PostScript are ok; Word, TEX or LATEX source are not), including:
   (a) a description of your designs and implementations for Tasks 1-4, explaining your choices;
   (b) a plot of the protocol running time vs p;
   (c) the successful response t for the attack in Task 2 with the parameter and observed pairs values in your dataXxxxx.txt file;
   (d) the success probability of a single attempt and a plot of the success probability vs the number of attempts n for your attack in Task 3;
   (e) the successful response t for the attack in Task 4 with the parameter values in your dataXxxxx.txt file.