Esecurity: secure internet & e-voting, summer 2013

MICHAEL NÜSKEN

3. Exercise sheet
Hand in solutions until Monday, 29 April 2013, 10:00

Exercise 3.1. (0 points)

-

Exercise 3.2 (X.509). (8 points)

Read RFC 5280 and answer the following questions:

(i) What classes of certificates are there? $\boxed{2}$

(ii) What is the basic syntax of X.509 v3 certificates? Describe the `Certificate Fields` in detail. Which signature algorithms are supported? $\boxed{2}$

(iii) What is a trust anchor? Can one use different trust anchors? $\boxed{2}$

(iv) What conditions are satisfied by a prospective certification path in the path validation process? $\boxed{2}$

Exercise 3.3 (Security estimate). (0+8 points)

RSA is a public-key encryption scheme that can also be used for generating signatures. It is necessary for its security that it is difficult to factor large numbers (which are a product of two primes). The best known factoring algorithms achieve the following (heuristic, expected) running times:

method	year	time for n-bit integers
trial division	$-\infty$	$\mathcal{O}^{\sim}\left(2^{n/2}\right)$
Pollard's $p-1$ method	1974	$\mathcal{O}^{\sim}\left(2^{n/4}\right)$
Pollard's ϱ method	1975	$\mathcal{O}^{\sim}\left(2^{n/4}\right)$
Pollard's and Strassen's method	1976	$\mathcal{O}^{\sim}\left(2^{n/4}\right)$
Morrison's and Brillhart's continued fractions	1975	$2^{\mathcal{O}(1) n^{1/2} \log_2^{1/2} n}$
Dixon's random squares	1981	$2^{(\sqrt{2}+o(1)) n^{1/2} \log_2^{1/2} n}$
Lenstra's elliptic curves method	1987	$2^{(1+o(1)) n^{1/2} \log_2^{1/2} n}$
quadratic sieve		$2^{(1+o(1)) n^{1/2} \log_2^{1/2} n}$
general number field sieve	1990	$2^{((64/9)^{1/3}+o(1)) n^{1/3} \log_2^{2/3} n}$

It is not correct to think of $o(1)$ as zero, but for the following rough estimates just do it, instead add a $\mathcal{O}(1)$ factor. Factoring the 768-bit integer RSA-768 needed about 1500 2.2 GHz CPU years (ie. 1500 years on a single 2.2 GHz AMD CPU) using the general number field sieve. Estimate the time that would be needed to factor an n-bit RSA number assuming the above estimates are accurate with $o(1) = 0$ (which is wrong in practice!)

+1 (i) for $n = 1024$ (standard RSA),

+1 (ii) for $n = 2048$ (as required for Document Signer CA),

+1 (iii) for $n = 3072$ (as required for Country Signing CA).

+2 (iv) Now assume that the attacker has 1000 times as many computers and 1000 times as much time as in the factoring record. Which n should I choose to be just safe from this attacker?

Repeat the estimate assuming that only Pollard's ϱ method is available

+1 (v) for $n = 1024$,

+1 (vi) for $n = 2048$,

+1 (vii) for $n = 3072$.

Remark: The statistics for discrete logarithm algorithms are somewhat similar as long as we consider groups $\mathbb{Z}_p^{\times}$. For elliptic curves (usually) only generic algorithms are available with running time $2^{n/2}$.