Esecurity: secure internet & e-passports, summer 2014

MICHAEL NÜSKEN

2. Exercise sheet
Hand in solutions until Monday, 21 April 2014, 23:59

Exercise 2.1 (GnuPG). (10 points)

(i) Which cryptographic algorithms are implemented in GnuPG? How is the idea of a hybrid crypto system implemented in GnuPG? 3

(ii) Read PHONG Q. NGUYEN, *Can We Trust Cryptographic Software? Cryptographic Flaws in GNU Privacy Guard v1.2.3*. How does the used implementation for RSA differ from the textbook version? What are the consequences? 3

(iii) Consider the model of trust in GnuPG. Describe how trust is transfered (ie. which keys are trusted?). Which parameters can be adjusted? 4

Exercise 2.2 (Hybrid crypto). (14+2 points)

Consider the situation in the exercises 1.2 and 1.3 from the last sheet. Eve has eavesdropped the conversation between Alice and Bob. She has recorded the RSA-cypher text $c = \mathrm{enc}_{(N,e)}(k)$ of the AES key k. She tries the following attack to recover k from c. We consider an attack as successful if it takes less than 2^{100} bit operations.

(i) How could Eve recover k if she tries all possible values? Is this a successful attack? 2

(ii) Eve computes $cx^{-e} \bmod N$ and y^e for all $1 \le x, y \le 2^{64}$ and stores these values in two lists. How can Eve recover k from these lists? Is this a successful attack? 4

(iii) The attack in (ii) may fail in some situations. In which does it fail? What is the probability of failing? 2+2

(iv) Eve finds that $e = 3$. Can she successfully recover k even if the attack in (ii) fails? 3

(v) How can one fix the vulnerability in the way RSA and AES is employed by Alice and Bob? 3

Exercise 2.3 (Security estimate). (0+5 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?

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 $2^{n/2}$.