3. Exercise sheet
Hand in solutions until Monday, 25 April 2011, 23:59

Exercise 3.1 (GnuPG). (6 points)

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

(ii) Which cryptographic algorithms are implemented in GnuPG? 2

Exercise 3.2 (X.509). (10 points)

Read RFC 5280 and answer the following questions:

(i) What classes of certificates are there? 2

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

(iii) What format has the Serial Number? What kind of knowledge do you gain from the Serial Number? 2

(iv) What is a trust anchor? Can one use different trust anchors? 2

(v) What conditions are satisfied by a prospective certification path in the path validation process? 2
Exercise 3.3 (Security estimate). (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:

<table>
<thead>
<tr>
<th>method</th>
<th>year</th>
<th>time for n-bit integers</th>
</tr>
</thead>
<tbody>
<tr>
<td>trial division</td>
<td>$-\infty$</td>
<td>$O^\sim{(2^{n/2})}$</td>
</tr>
<tr>
<td>Pollard’s $p-1$ method</td>
<td>1974</td>
<td>$O^\sim{(2^{n/4})}$</td>
</tr>
<tr>
<td>Pollard’s $\varrho$ method</td>
<td>1975</td>
<td>$O^\sim{(2^{n/4})}$</td>
</tr>
<tr>
<td>Pollard’s and Strassen’s method</td>
<td>1976</td>
<td>$O^\sim{(2^{n/4})}$</td>
</tr>
<tr>
<td>Morrison’s and Brillhart’s continued fractions</td>
<td>1975</td>
<td>$2^{O(1)n^{1/2}\log^{1/2} n}$</td>
</tr>
<tr>
<td>Dixon’s random squares</td>
<td>1981</td>
<td>$2^{(\sqrt{2}+o(1))n^{1/2}\log^{1/2} n}$</td>
</tr>
<tr>
<td>Lenstra’s elliptic curves method</td>
<td>1987</td>
<td>$2^{(1+o(1))n^{1/2}\log^{1/2} n}$</td>
</tr>
<tr>
<td>quadratic sieve</td>
<td></td>
<td>$2^{(1+o(1))n^{1/2}\log^{1/2} n}$</td>
</tr>
<tr>
<td>general number field sieve</td>
<td>1990</td>
<td>$2^{((64/9)^{1/3}+o(1))n^{1/3}\log^{2/3} n}$</td>
</tr>
</tbody>
</table>

It is not correct to think of $o(1)$ as zero, but for the following rough estimates just do it, instead add a $O(1)$ 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!)

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

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

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

(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 $\varrho$ method is available

(v) for $n = 1024$, 
(vi) for $n = 2048$,

(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 $2^{n/2}$.

**Exercise 3.4** (Dixon’s random squares). (0+4 points)

(i) Let $N = q_1 q_2 \cdots q_r$ be odd with pairwise distinct prime divisors $q_i$ and $r \geq 2$. Show: The equation $x^2 - 1 = 0$ has exactly $2^r$ solutions in $\mathbb{Z}_N^\times$.

*Hint:* Use the Chinese remainder theorem.

*Note:* The claim is also true, if the $q_i$ are pairwise distinct prime powers. To see this you have to know that also for prime powers $q$ the equation $x^2 - 1 = 0$ has exactly 2 solutions in $\mathbb{Z}_q$.

(ii) If $s, t$ are random elements of $\mathbb{Z}_N^\times$ satisfying $s^2 \equiv t^2 \mod N$, then the probability for $s \not\equiv \pm t \mod N$ is at least $1 - \frac{1}{2^r}$.