Cryptography (SS2006) Prof. Michael Backes

Solutions for Exercise Sheet 10

Out: 07/13/2006 Saarland University

Problem 1: Square Roots modulo Composites

By the Chinese Reminder Theorem we can compute the square root of 58 modulo the prime factors of 77, namely 7 and 11, and then compute the root modulo 77 from these. Thus first we calculate both roots modulo 7

$$58 = 2 \bmod 7 \text{ and } y_7^{(1)/(2)} := \pm 2^{\frac{7+1}{4}} = \pm 2^2 = \pm 4 \bmod 7$$

and both roots modulo 11

$$58 = 3 \bmod 11 \text{ and } y_{11}^{(1)/(2)} := \pm 3^{\frac{11+1}{4}} = \pm 3^3 = \pm 3 \cdot 9 = \pm 5 \bmod 11.$$

Now, using the extended euclidean algorithm, we find that

$$2 \cdot 11 - 3 \cdot 7 = 22 - 21 = 1,$$

thus we obtain the four square roots modulo 58 as

$$\begin{aligned}
y^{(1)} &:= 2 \cdot 11 \cdot y_7 - 3 \cdot 7 \cdot y_{11} \\
&= 2 \cdot 11 \cdot 4 - 3 \cdot 7 \cdot 5 \\
&= 88 - 105 = -17 = 60 \bmod 77, \\
y^{(2)} &:= 2 \cdot 11 \cdot (-y_7) - 3 \cdot 7 \cdot y_{11} \\
&= -88 - 105 = 38 \\
y^{(3)} &:= 2 \cdot 11 \cdot y_7 - 3 \cdot 7 \cdot (-y_{11}) \\
&= 88 + 105 = 39 \\
y^{(4)} &:= 2 \cdot 11 \cdot (-y_7) - 3 \cdot 7 \cdot (-y_{11}) \\
&= -88 + 105 = 17
\end{aligned}$$

and indeed

$$60^2 = 38^2 = 39^2 = 17^2 = 58 \bmod 77.$$

Problem 2: Factoring and Computing Square Roots

Suppose we have access to an algorithm F which, on input a quadratic residue $y \in \mathbb{Z}_N$, outputs an x such that $x^2 = y \bmod N$ with probability ϵ.

We construct the following algorithm. Choose a random $a \in \mathbb{Z}_N$, compute $d := a^2$, and run the algorithm $\mathsf{F}(d)$, which finally outputs an an $b \in \mathbb{Z}_N$ with $b^2 = d$ with probability ϵ. If $b = a$ or $b = -a$ we repeat this step, otherwise we proceed. (Note that there are four square roots for d, and two of them are "good" ones. As F does not get any information which square root we already know, we have probability $1/2$ to get a "good" square root.)

By construction

$$a^2 = b^2 \bmod N,$$

thus $N \mid a^2 - b^2$. This difference can be rewritten as $(a+b)(a-b)$, thus

$$N \mid (a-b)(a+b).$$

But $N \not\mid (a-b)$, as $-N < (a-b) < N$, and $N \not\mid (a+b)$, as $0 < (a+b) < 2N$ and $(a+b) \neq N$ (otherwise $a = -b$).

Thus the two factors p and q of N are "distributed" across these two factors $(a+b)$ and $(a-b)$, thus $\gcd((a-b), N)$ is a non-trivial factor of N (and $\gcd((a+b), N)$ is the other).

Problem 3: Commitment Schemes

(a) It is obvious that the scheme is *correct*.

However, in general it is neither binding nor hiding. Let $\mathsf{H}' : \{0,1\}^n \to \{0,1\}^n$ be a OWF, then

$$\mathsf{H} : \{0,1\}^{n+1} \to \{0,1\}^{n+1}, (a,b) \mapsto (\mathsf{H}(a), b)$$

where $a \in \{0,1\}^n$ and $b \in \{0,1\}$ is a OWF as well. The described scheme using this one-way function H is obviously neither binding nor hiding.

(b) The trick is to encode the bit b as a hard-core predicate for the concrete one-way function. Remember that for the discrete exponentiation function $\mathbb{Z}_p \to \mathbb{Z}_p^*,\ x \mapsto g^x$, the information if $x \geq \frac{p-1}{2}$ or $x < \frac{p-1}{2}$ is a hard-core predicate, c.f. homework sheet 7.

Thus, to commit to a bit b, we choose $r \leftarrow_{\mathcal{R}} \{1, \ldots, \frac{p-1}{2}\}$ and let $h := g^{b \cdot \frac{p-1}{2} + r}$ be the commitment to the value b. To open this commitment he sends b and r to the verifier, who recomputes and verifies h.

This scheme is obviously correct. It is also information-theoretical binding, as g is a generator and thus there exists a bijection between exponents (in the range $0, \ldots, p$) and group elements. The hiding property follows directly from the fact that b has the same value as the hard-core predicate, which is infeasible, if the DLog is hard.

Problem 4: Commitment Schemes II

(i) If the committer does not verify that p, q are primes, then the receiver can use $p = 2^n + 1$ and $q = 2^k$. This means that one can compute discrete logarithms in $\mathbb{Z}_p^*$ quite efficiently as we have seen on the exercise sheet 5. Thus the committer can compute x such that $g = h^x$, and consequently can break the binding property: For arbitrary m, r, r' let $m' := m + xr - xr'$, then the commitment $g^m h^r$ can be opened with both m, r and m', r'.

Note that this weakness can be seen as esoteric, as the committer can cheat a verifier that is cheating itself. However, in practice it can be the case that the values p, q, g, h are chosen by a central authority and then published. So this problem teaches us that such a central authority really needs to be trusted.

(ii) If the committer does not verify that h has the order q, then the verifier can send, e.g., $h = 1$. This means that the commitment to a message m is computed as $g^m h^r$ for a random element r. But if $h = 1$, this term falls out, thus the commitment becomes g^m. This, however, does not have the hiding property any more, as an attacker can distinguish different messages m_0, m_1 by computing g^{m_0} and g^{m_1}.