Cryptography (SS2006) Prof. Michael Backes

Solutions for Exercise Sheet 5

Out: 06/08/2006 Saarland University

Problem 1: Discrete Logarithm in $\mathbb{Z}^*_{2^n+1}$

Let $p = 2^n + 1$ a prime, g a generator of $\mathbb{Z}^*_p$, and $h \in \mathbb{Z}^*_p$. We write $\mathsf{DLog}_g(h) = x = \sum_{i=1}^{n-1} x_i 2^i$. Then we calculate as follows:

$$\begin{aligned} h^{2^{n-1}} = g^{2^{n-1}} \bmod p \quad &\Leftrightarrow \quad g^{x\cdot 2^{n-1}} = g^{2^{n-1}} \bmod p \\ &\Leftrightarrow \quad x \cdot 2^{n-1} = 2^{n-1} \bmod 2^n \end{aligned}$$

where the last equivalence is true by the hint we gave. Now we calculate the left expression as

$$\begin{aligned} x \cdot 2^{n-1} &= \left(\sum_{i=0}^{n-1} x_i \cdot 2^i\right) \cdot 2^{n-1} \\ &= \left(\sum_{i=1}^{n-1} x_i \cdot 2^{n-1+i}\right) + x_0 \cdot 2^{n-1} \\ &= x_0 \cdot 2^{n-1} \bmod 2^n. \end{aligned}$$

The last equality holds as we are calculating module 2^n. Then it follows

$$\begin{aligned} x_0 \cdot 2^{n-1} \bmod 2^n = 2^{n-1} \bmod 2^n & \\ \Leftrightarrow \quad (x_0 - 1) \cdot 2^{n-1} = 0 \bmod 2^n & \\ \Leftrightarrow \quad x_0 = 1 & \end{aligned}$$

A similar calculation shows

$$h^{2^{n-2}} = g^{2^{n-1}+x_0\cdot 2^{n-2}} \bmod p.$$

Again, calculating in the exponent, this is equivalent to

$$x \cdot 2^{n-2} = 2^{n-1} + x_0 \cdot 2^{n-2} \bmod 2^n,$$

and this can be simplified to

$$x_1 \cdot 2^{n-1} + x_0 \cdot 2^{n-2} = 2^{n-1} + x_0 \cdot 2^{n-2} \bmod 2^n$$

Finally, this is equivalent to

$$x_1 = 1$$

But now the general construction should be rather clear. For $k = 2, \ldots, n-1$ we subsequently calculate x_i as

$$x_k = 1 \quad \text{iff} \quad h^{2^{n-1-k}} = g^{2^{n-1}+(\sum_{j=0}^{k-1} x_j\cdot 2^j)\cdot 2^{n-1-k}} \bmod p$$

A similar calculation shows that

$$\begin{aligned} \Leftrightarrow h^{2^{n-1-k}} &= g^{2^{n-1}+(\sum_{j=0}^{k-1} x_j\cdot 2^j)\cdot 2^{n-1-k}} \bmod p \\ \Leftrightarrow x \cdot 2^{n-1-k} &= 2^{n-1} + (\sum_{j=0}^{k-1} x_j \cdot 2^j) \cdot 2^{n-1-k} \bmod 2^n \\ \Leftrightarrow (\sum_{j=0}^{k} x_j \cdot 2^j) \cdot 2^{n-1-k} &= 2^{n-1} + (\sum_{j=0}^{k-1} x_j \cdot 2^j) \cdot 2^{n-1-k} \bmod 2^n \\ \Leftrightarrow x_k \cdot 2^{n-1} &= 2^{n-1} \bmod 2^n \\ \Leftrightarrow x_k &= 1 \bmod 2^n \end{aligned}$$

Problem 2: Random Self-reducibility of CDH

Claim: Let p be a prime and g be a generator of $\mathbb{Z}_p^*$. We suppose that there exists an algorithm A for computing g^{xy} in time T whenever $(g^x, g^y) \in S_1 \times S_2$, where $S_1, S_2 \subset \mathbb{Z}_p^*$ are arbitrary sets with $|S_1| = |S_2| = \epsilon \cdot |\mathbb{Z}_p^*|$. Then there exists an algorithms B that computes g^{xy} for all $(g^x, g^y) \in \mathbb{Z}_p^* \times \mathbb{Z}_p^*$ in expected time T/ϵ^2.

Proof. We are given two arbitrary elements $g^x, g^y \in \mathbb{Z}_p^*$ and a generator g of $\mathbb{Z}_p^*$. We show the existence of the algorithm B by constructing it.

We pick two elements $u \leftarrow_{\mathcal{R}} \mathbb{Z}_p^*$, $v \leftarrow_{\mathcal{R}} \mathbb{Z}_p^*$ at random and compute g^u and g^v, which are also uniformly distributed over $\mathbb{Z}_p^*$ as g is a generator of $\mathbb{Z}_p^*$. Now we let A compute $\mathsf{A}(g^u g^x, g^v g^y) = \mathsf{A}(g^{u+x}, g^{v+y})$. If A returns the distinguished error element $\downarrow$, we pick again two elements $u \leftarrow_{\mathcal{R}} \mathbb{Z}_p^*$, $v \leftarrow_{\mathcal{R}} \mathbb{Z}_p^*$ at random and test again whether $\mathsf{A}(g^{u+x}, g^{v+y})$ returns $\downarrow$ or not. If A does not return $\downarrow$, we know that

$$\mathsf{A}(g^{u+x}, g^{v+y}) = g^{((u+x)(v+y))} = g^{uv+xv+uy+xy} = g^{xy} \cdot (g^{uv} \cdot g^{xv} \cdot g^{uy}).$$

Hence we compute g^{xy} as

$$g^{xy} = \frac{g^{xy} \cdot (g^{uv} \cdot g^{xv} \cdot g^{uy})}{g^{uv} \cdot g^{xv} \cdot g^{uy}}.$$

We can compute this as $g^{xv} = (g^x)^v$, $g^{uy} = (g^y)^u$ and g, u, v are known to us.

Now we look at the expected runtime of the algorithm B. Since g^u and g^v are uniformly drawn from $\mathbb{Z}_p^*$, the probability of g^{u+x} being in S_1 and g^{v+y} being in S_2 is ϵ, respectively. Hence the probability of the pair (g^{u+x}, g^{v+y}) being in $S_1 \times S_2 \subset \mathbb{Z}_p^* \times \mathbb{Z}_p^*$ is $\epsilon \cdot \epsilon = \epsilon^2$. The algorithm B takes an expected number of $\frac{1}{\epsilon^2}$ loops, each of which takes time T for the algorithm A, thus the expected runtime is T/ϵ^2. ∎

Problem 3: Cyclic Groups

(a)

"$\Rightarrow$" Let g be a generator of $\mathbb{Z}_p^*$, i.e., $\langle g \rangle = \left\{g^0, g^1, \ldots, g^{p-2}\right\} = \mathbb{Z}_p^*$. Thus, $g^i \neq g^j \mod p, \forall i \neq j$ where $0 \leq i, j \leq p-2$. In particular, $g^q \neq g^0 = 1 \mod p$ as $0 < q \leq p-2$ for $q \geq 1$, and $g^2 \neq g^0 = 1 \mod p$ as $0 < 2 \leq p-2$. Note that p and q are prime and $p = 2q+1$, thus $p \geq 4$.

"$\Leftarrow$" Let $g \in \mathbb{Z}_p^*$ such that $g^q \neq 1 \mod p$ and $g^2 \neq 1 \mod p$. By Lagranges Theorem we also have that $ord_p(g) \mid p-1 = 2q$. As 2 and q are prime numbers $ord_p(g) \in \{1, 2, q, 2q\}$. We do case analysis on the value of $ord_p(g)$:

Case 1: $ord_p(g) = 1$. This implies that $g = g^1 = 1 \mod p$ and thus $g^2 = 1$, which contradicts $g^2 \neq 1$.

Case 2: $ord_p(g) = 2$. This implies that $g^2 = 1 \mod p$ and contradicts $g^2 \neq 1 \mod p$.

Case 3: $ord_p(g) = q$. This implies that $g^q = 1 \mod p$ and contradicts $g^q \neq 1 \mod p$.

Case 4: Since we obtained contradictions in all the other cases, the only possible value for $ord_p(g)$ is $2q = p-1$. Thus g is a generator of $\mathbb{Z}_p^*$, which concludes our proof.

(b) Let $g \in \mathbb{Z}_p^*$ such that $\langle g \rangle = \mathbb{Z}_p^*$ and let $i \in N$ such that $2 \nmid i$ and $q \nmid i$. Since for $i = 1$ the conclusion holds trivially we only consider $i \neq 1$. We use (a), so it is sufficient to prove that $g^{qi} \neq 1 \mod p$ and $g^{2i} \neq 1 \mod p$.

To get a contradiction we assume that $g^{qi} = 1 \mod p$. Then we have $p-1 = ord_p(g) \mid qi$, this implies $2q \mid qi$, thus $2 \mid i$ which contradicts $2 \nmid i$.

Similarly, we assume that $g^{2i} = 1 \mod p$. Then we have $p-1 = ord_p(g) \mid 2i$, which implies $2q \mid 2i$, thus $q \mid i$ which contradicts $q \nmid i$.

So we proved that $g^{qi} \neq 1 \mod p$ and $g^{2i} \neq 1 \mod p$, which using (a) is equivalent to $\left\langle g^i \right\rangle = \mathbb{Z}_p^*$.

Problem 4: Some Computations

(a) There are different ways to compute the inverse of an element x, one is to compute x^{p-2}. By Fermat we have $x \cdot x^{p-2} = x^{p-1} = 1 \bmod p$. Thus we can compute

$$\begin{aligned} 30^{-1} = 30^{71} &= 30^{64} \cdot 30^4 \cdot 30^2 \cdot 30^1 \\ &= 24^{32} \cdot 24^2 \cdot 24^1 \cdot 30^1 \\ &= 65^{16} \cdot 65^1 \cdot 24^1 \cdot 30^1 \\ &= 64^8 \cdot 65^1 \cdot 24^1 \cdot 30^1 \\ &= 8^4 \cdot 65^1 \cdot 24^1 \cdot 30^1 \\ &= 64^2 \cdot 65^1 \cdot 24^1 \cdot 30^1 \\ &= 8^1 \cdot 65^1 \cdot 24^1 \cdot 30^1 \\ &= 56 \bmod 73 \end{aligned}$$

(b) $p = 19 = 3 \bmod 4$, so we can use exponentiation with $\frac{p+1}{4}$ to compute square roots. So

$$\sqrt{11} = 11^{\frac{19+1}{4}} = 11^5 = 11 \cdot 11^4 = 11 \cdot 7^2 = 11 \cdot 11 = 7 \bmod 19.$$

(c) Again $p = 11 = 3 \bmod 4$ so

$$\sqrt{5} = 5^3 = 5 \cdot 5^2 = 5 \cdot 3 = 4 \bmod 11.$$

(d)

$$\begin{aligned} 3^{42} &= 3^{32} \cdot 3^8 \cdot 3^2 \\ &= 9^{16} \cdot 9^4 \cdot 9^1 \\ &= 81^8 \cdot 81^2 \cdot 9^1 \\ &= 97^4 \cdot 97^1 \cdot 9^1 \\ &= 16^2 \cdot 97^1 \cdot 9^1 \\ &= 54^1 \cdot 97^1 \cdot 9^1 \\ &= 76 \bmod 101 \end{aligned}$$

(e)

$$\begin{aligned} 7^{23} &= 7^{16} \cdot 7^4 \cdot 7^2 \cdot 7^1 \\ &= 49^8 \cdot 49^2 \cdot 49^1 \cdot 7^1 \\ &= 47^4 \cdot 47^1 \cdot 49^1 \cdot 7^1 \\ &= 69^2 \cdot 47^1 \cdot 49^1 \cdot 7^1 \\ &= 53^1 \cdot 47^1 \cdot 49^1 \cdot 7^1 \\ &= 18 \bmod 107 \end{aligned}$$