CS174 Sp2001 **Homework 12 Solutions** **out: May 3, 2001**

1. Each secret share s_i of a secret s is a pair x_i, y_i where $y_i = p(x_i)$ and

$$p(x) = r_t x^t + \cdots + r_1 x + s (\mathrm{mod}\ p)$$

is a polynomial whose coefficients $r_1, \ldots, r_t$ are chosen independently and uniformly at random from $\mathbb{Z}_p$. As we did for addition, assume that all secrets are shared at the same evaluation points $x_1, \ldots, x_n$. Then we can drop references to the x_i, and write (by slight abuse of notation) $s_i = y_i$.

Suppose then that secrets a and b are shared as $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$. The reconstruction function h is:

$$s = h(s_1, \ldots, s_{t+1}) = \sum_{i=1}^{t+1} s_i L_i$$

and the Lagrange polynomial coefficient L_i (which we wrote as $L_i(0)$ before) is

$$L_i = \frac{\prod_{j \neq i} -x_j}{\prod_{j \neq i} (x_i - x_j)}$$

the question asks to compare the values of $h(a_1 b_1, \ldots, a_{t+1} b_{t+1})$ and ab, and from the above:

$$h(a_1 b_1, \ldots, a_{t+1} b_{t+1}) = \sum_{i=1}^{t+1} a_i b_i L_i$$

to prove that this is not equal to ab we need only a counter-example. Pick $t = 1$, $x_1 = -1$, $x_2 = 1$, whence $L_1 = L_2 = 1/2$. Therefore $h(a_1 b_1, a_2 b_2) = 1/2(a_1 b_1 + a_2 b_2)$. But ab is the product of $1/2(a_1 + a_2)$ and $1/2(b_1 + b_2)$. Clearly:

$$1/2(a_1 b_1 + a_2 b_2) \neq 1/4(a_1 + a_2)(b_1 + b_2)$$

2. Notice that $h(s_1, \ldots, s_{t+1})$ is a linear function from the formula above, that is, $h(\lambda s_1, \ldots, \lambda s_{t+1}) = \lambda h(s_1, \ldots, s_{t+1}) (\mathrm{mod}\ p)$. So $h(ka_1, \ldots, ka_{t+1}) = ka (\mathrm{mod}\ p)$. Therefore multiplication by a public scalar works with secret-sharing.

3. Let u and v be two numbers bit-committed as $A = g^x h^u$ and $B = g^y h^v$. We give two ZKPs, one that $(u = 1) \vee (v = 1)$ and the other that $(u = 0) \vee (v = 0)$. If both conditions hold, then exactly one of the numbers is zero, and the other is one. First, for the proof that $(u = 1) \vee (v = 1)$. In reality, it will be the case that either $u = 0$, $v = 1$ or vice versa. Suppose the first case holds, then we will need a simulation of a proof that $u = 1$ and a real proof that $v = 1$, and we will combine them:

   (a) Prover picks a_1 (for real proof that $v = 1$) at random and sends $\alpha_1 = g^{a_1} (\mathrm{mod}\ p)$ to verifier. Prover picks random c_0 and w_0 and sets $\alpha_0 = g^{w_0} (A h^{-1})^{-c_0} (\mathrm{mod}\ p)$, and sends α_0 to verifier (for phoney proof that $u = 1$).

(b) Verifier picks $c \pmod q$ at random, and sends it to prover.

(c) Prover computes $c_1 = c - c_0$, and then $w_1 = yc_1 + a_1 \pmod q$. Prover sends c_0, c_1, and w_0 and w_1 to verifier.

(d) Verifier checks that $c = c_0 + c_1$ and that

$$g^{w_0} = \alpha_0 (Ah^{-1})^{c_0} \pmod p$$
$$g^{w_1} = \alpha_1 (Bh^{-1})^{c_1} \pmod p$$

For the proof for the case where $u = 1$ and $v = 0$ is similar, we flip the correct and phoney proofs:

(a) Prover picks a_0 (for real proof that $u = 1$) at random and sends $\alpha_0 = g^{a_0} \pmod p$ to verifier. Prover picks random c_1 and w_1 and sets $\alpha_1 = g^{w_1}(Ah^{-1})^{-c_1} \pmod p$, and sends α_1 to verifier (for phoney proof that $v = 1$).

(b) Verifier picks $c \pmod q$ at random, and sends it to prover.

(c) Prover computes $c_0 = c - c_1$, and then $w_0 = yc_0 + a_0 \pmod q$. Prover sends c_0, c_1, and w_0 and w_1 to verifier.

(d) Verifier checks that $c = c_0 + c_1$ and that

$$g^{w_0} = \alpha_0 (Ah^{-1})^{c_0} \pmod p$$
$$g^{w_1} = \alpha_1 (Bh^{-1})^{c_1} \pmod p$$

To construct a proof that $u = 0$ or $v = 0$, we repeat the above proofs, but replace (Ah^{-1}) with (A) and (Bh^{-1}) with (B).

Second Method This method is a little simpler. Notice that if exactly one of u, v is one and the other zero, then $u + v = 1$. Use the proof given in class to show that u is either zero or one. Then by enforcing the constraint that $u + v = 1$, we force v to be either zero or one. To enforce the constraint, note that

$$ABh^{-1} = g^x h^u g^y h^v h^{-1} = g^{(x+y)} h^{(u+v-1)}$$

and then we can give a zero-knowledge proof that we know the discrete log wrt g of ABh^{-1}. That proves that ABh^{-1} is a pure power of g (assuming we dont know the log of h), or in other words $u + v - 1 = 0$. This proof is just Shamir's discrete log proof:

(a) Prover picks a at random, and sends $\alpha = g^a \pmod p$ to verifier.

(b) Verifier picks c at random from $\mathbb{Z}_q$ and sends to prover.

(c) Prover sends $w = c(x + y) + a \pmod q$ to verifier.

(d) Verifier checks that $g^w = \alpha(ABh^{-1})^c \pmod p$.