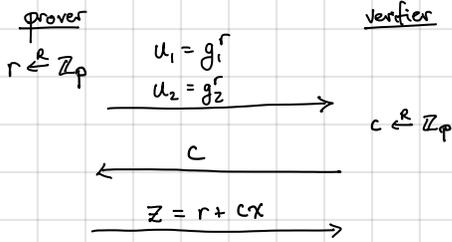


Starting point: proof of knowledge of two discrete logs (for fixed  $g_1, g_2$ )

$$\mathcal{L} = \{(u, v) \in \mathbb{G} : \exists r \in \mathbb{Z}_p : u = g_1^r, v = g_2^r\}$$



Check that  $g_1^z = u_1 \cdot h_1^c$  and  $g_2^z = u_2 \cdot h_2^c$

Completeness and HVZK follows as in Schnorr's protocol.

Knowledge: Two scenarios:

- If prover uses inconsistent commitment (i.e.,  $u_1 = g_1^{r_1}$  and  $u_2 = g_2^{r_2}$  where  $r_1 \neq r_2$ ), then over choice of honest verifier's randomness, then prover can only succeed with probability at most  $1/p$ :

$$z = r_1 + x_1 c = r_2 + x_2 c \quad (\text{if verifier accepts})$$

$$u_1 = g_1^{r_1} \quad h_1 = g_1^{x_1} \quad u_2 = g_2^{r_2} \quad h_2 = g_2^{x_2}$$

This means that

$$(r_1 - r_2) = t(x_2 - x_1)$$

If  $r_1 \neq r_2$ , there is at most 1  $c \in \mathbb{Z}_p$  where this relation holds. Since  $c$  is uniform over  $\mathbb{Z}_p$ , the verifier accepts with probability at most  $1/p$

- If prover succeeds with  $1/\text{poly}(k)$  probability, then it must use a "consistent" commitment. Can build extractor as in Schnorr's protocol. Knowledge error larger by additive  $1/p$  term (from above analysis).

Our language of valid votes:

$$\mathcal{L} = \{(u, v) : \exists r : (u = g_1^r, v = g_2^r \text{ or } u = g_1^r, v = g_2^r \cdot g_1)\}$$

Equivalently: either know  $r$  such that

$$u = g_1^r, v = g_2^r \quad \text{or} \quad u = g_1^r, \sqrt{v/g_1} = g_2^r$$

Looks like statement for knowledge of two discrete logs  
(either for statement  $(u, v)$  or for statement  $(u, \sqrt{v/g_1})$ )

Or-proof: A general approach for proving or of two statements (without revealing which one is true)

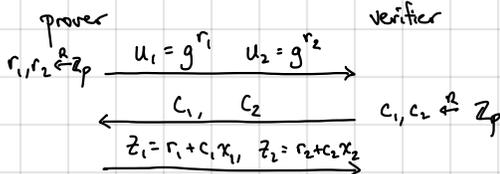
We will illustrate for simple case of

$$\mathcal{L} = \{(h_1, h_2) : \exists x (h_1 = g_1^x \text{ or } h_2 = g_2^x)\} \quad [\text{for fixed generator } g]$$

Prover demonstrates knowledge of discrete log of either  $h_1$  or  $h_2$

Note: prover may only know one of  $x_1, x_2$

Starting point: Run two copies of Schnorr to prove knowledge of  $(r_1, r_2)$  such that  $h_1 = g_1^{x_1}$  and  $h_2 = g_2^{x_2}$

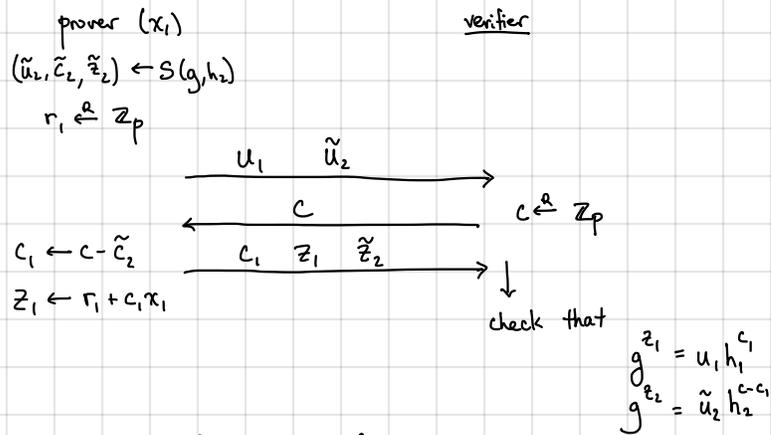


Problem: Honest prover only knows one of  $x_1$ , or  $x_2$  so it cannot correctly answer both challenges (unless it knew both  $x_1$  and  $x_2$ )

Key idea: Prover will simulate the transcript it does not know.

Suppose prover knows  $x = x_1$ . Then, it will first run the Schnorr simulator on input  $(g, h_2)$  to obtain transcript  $(\tilde{u}_2, \tilde{c}_2, \tilde{z}_2)$ .

↳ But challenge  $c_2$  may not match  $\tilde{c}_2$ ... To address this, we will have the verifier send a single challenge  $c \in \mathbb{Z}_p$  and the prover can pick  $c_1$  and  $c_2$  such that  $c_1 + c_2 = c \in \mathbb{Z}_p$



Completeness, HVZK and proof of knowledge follow very similarly as in the proof of Schnorr's protocol

Proving that  $(u, v)$  have the form  $(u, v) = (g^u, h^v)$  or  $(u, v/g) = (g^u, h^v)$  can be done by combining or-proof with proof of knowledge of two discrete logs described above.

- Namely, prover simulates proof of instance that is false and proves the statement that is true