0.1 Sigma(Σ) Protocol

Typically, a zero knowledge proof protocol involves a prover P, a verifier V, and a binary relation $R \subset \{0,1\}^* \times \{0,1\}^*$. Suppose $(x, w) \in R$, x is an instance of a computational problem, and w is the solution (witness) to the instance. For the discrete log problem, the relation R_{DL} is defined as $R_{DL} = \{((G, p, q, h), w) | h = g^w\}$. Suppose P wants to prove that he knows a witness w for which $(x, w) \in R$ without revealing anything. If the prove protocol is a three-round public-coin protocol, then we say it is a Σ protocol. One typical example of Σ protocol is Schnorr's protocol for discrete log, which is presented as follows. Suppose G is a group of order q, with the generator g. P and V have the input $h \in G$, and P proves that he knows a secret witness w, such that $g^w = h$.

1. P chooses a random number $r \in Z_q$, computes $a = g^r \bmod p$, and sends a to V;
2. V choose a random challenger $e \leftarrow \{0,1\}^t$, t is a fixed number that $2^t < q$. V sends e to P.
3. P computes $z = r + ew \bmod q$, and sends z to V.
4. V checks if the following equation holds: $g^z = ah^e \bmod q$

0.1.1 Non-interactive Σ Protocol

Interactive $Sigma$ protocols can be converted to a non-interactive protocol using the Fiat-Shamir transform. The idea is that instead of receiving e from the verifier, the prover compute the value of a hash function on the first message(a) and the input(h). Hence the non-interactive Schnorr procotol becomes as follows.

1. P chooses a random number $r \in Z_q$, computes $a = g^r \bmod p$, computes $e = H(a, g, h)$, computes $z = r + ew \bmod q$, and sends the proof $\pi = (a, e, z)$ to V.

2. On receiving the proof $\pi = (a, e, z)$, V checks the following two equations: $e = H(a, g, h)$, $g^z = ah^e$

0.1.2 Σ Protocol for a DH Tuple

Another useful example of *Sigma* protocol is the protocol for a Diffie-Hellman tuple. Suppose, P wants to prove to V that he knows a witness w such that $u = g^w$ and $v = h^w$.

1. P chooses a random number $r \in Z_q$, computes $a = g^r \bmod p$, $b = h^r \bmod p$ and sends a, b to V;
2. V choose a random challenger $e \leftarrow \{0, 1\}^t$, t is a fixed number that $2^t < q$. V sends e to P.
3. P computes $z = r + ew \bmod q$, and sends z to V.
4. V checks if the following equations hold: $g^z = au^e \bmod q$, $h^z = bvc^e \bmod q$

0.1.3 AND composition

The Σ protocol can be performed in parallel to prove the **AND** pf multiple statements. The idea is to use the same challenger e for all statements. Suppose P wants to prove the knowledge of w_1, w_2, such that $h_1 = g^{w_1}, h_2 = g^{w_2}$. g, h_1, h_2 are public.

1. P chooses two random number $r_1, r_2 \in Z_q$, computes $a_1 = g^{r_1}, a_2 = g^{r_2} \bmod p$, and sends a_1, a_2 to V.
2. V choose a random challenger $e \leftarrow \{0, 1\}^t$, t is a fixed number that $2^t < q$. V sends e to P.
3. P computes $z_1 = r_1 + ew_1 \bmod q$, $z_2 = r_2 + ew_2 \bmod q$, and sends z_1, z_2 to V.

4. V checks if the following equations hold: $g^{z_1} = a_1 {h_1}^{e_1} \bmod q$, $g^{z_2} = a_2 {h_2}^{e_2} \bmod q$

0.1.4 OR composition

OR composition means that P wants to prove the knowledge of (at least) one of w_1, w_2, such that $h_1 = g^{w_1} h_2 = g^{w_2}$ without revealing which. Suppose P knows a witness w_1 for h_1. The idea is to generate a real proof of knowledge for w_1, but create a simulated proof for w_2.

1. P chooses a random number $r_1 \in Z_q$, computes $a_1 = g^{r_1} \bmod p$. Then P chooses a random e_2 to get (a_2, e_2, z_2) through similation. Finally, P sends (a_1, a_2) to V.

2. V choose a random challenger $e \leftarrow \{0,1\}^t$, t is a fixed number that $2^t < q$. V sends e to P.

3. P replies with e_1, e_2, such that $e_1 = e \oplus e_2$, and also sends z_1, z_2 to V. Note that P already has z_2, and computes z_1.