0.1 Sigma Protocols

0.2 Schnorr's Identification Protocol

One very useful protocol for demonstration purposes is Schnorr's identification protocol. It is a simple and elegant protocol that allows one party to prove to another party that it knows a discrete logarithm of a given element.

Let us formalize it using theory above. Introduce the language

Suppose $\mathbb{G}$ is a cyclic group of prime order q with generator $g \in \mathbb{G}$. Suppose prover $\mathcal{P}$ has a secret key $\alpha \in \mathbb{Z}_q$ and the corresponding public key $u = g^\alpha \in \mathbb{G}$ and he wants to convince the verifier $\mathcal{V}$ that he knows α corresponding to the public key u.

Well, the easiest way how to proceed is simply giving α to $\mathcal{V}$, but this is obviously not what we want. Instead, the Schnorr protocol allows $\mathcal{P}$ to prove the knowledge of α without revealing it.

Let us finally describe the protocol. The schnorr identification protocol $\Pi_{\mathsf{Schnorr}} = (\mathsf{Gen}, \mathcal{P}, \mathcal{V})$ with a generation function Gen and prover $\mathcal{P}$ and verifier $\mathcal{V}$ is defined as follows:

- $\mathsf{Gen}(1^\lambda)$: As with most public-key cryptosystems, we take $\alpha \xleftarrow{R} \mathbb{Z}_q$ and $u \leftarrow g^\alpha$. We output the *verification key* as $\mathsf{vk} := u$, and the *secret key* as $\mathsf{sk} := \alpha$.
- The protocol between $(\mathcal{P}, \mathcal{V})$ is run as follows:
  - $\mathcal{P}$ computes $\alpha_T \leftarrow \mathbb{Z}_q, u_T \leftarrow g^{\alpha_T}$ and sends u_T to $\mathcal{V}$.
  - $\mathcal{V}$ sends a random challenge $c \xleftarrow{R} \mathbb{Z}_q$ to $\mathcal{P}$.
  - $\mathcal{P}$ computes $\alpha_C \leftarrow \alpha_T + \alpha c \in \mathbb{Z}_q$ and sends α_C to $\mathcal{V}$.
  - $\mathcal{V}$ accepts if $g^{\alpha_C} = u_T \cdot u^c$, otherwise it rejects.