How the Schnorr Protocol Works

The Schnorr protocol is based on the concept of discrete logarithms, the inverse of the exponentiation function in a group. Typically, the group is a finite field or an elliptic curve group.

Protocol Steps

1. The prover generates a random number r and computes the value $a = g^r$.
2. The prover sends the value a to the verifier.
3. The verifier randomly selects a challenge e and sends it to the prover.
4. The prover computes the value $z = r + eu$ and sends it to the verifier.
5. The verifier verifies that $g^z = ay^e$. If this equation holds, then the verifier is convinced that the prover knows the secret value u.

Zero-Knowledge Property

The Schnorr protocol is a zero-knowledge protocol, meaning the verifier learns nothing about the prover's secret value u from the interaction. This is because the prover's responses are always consistent with any value of u.

Different Implementations

There are two main ways to implement the Schnorr protocol:

Interactive Implementation

This is the original implementation where the prover and verifier exchange messages back and forth. It is secure but inefficient.

Non-interactive Implementation

This modification allows the prover to generate a proof that can be verified without further interaction with the verifier. It is more efficient but requires a stronger computational assumption to be secure.

Non-interactive Implementation of the Schnorr Protocol

The non-interactive implementation is based on the Fiat-Shamir heuristic, converting a zero-knowledge protocol into a non-interactive protocol using a random oracle.

1. The prover generates a random number r and computes $a = g^r$.

2. The prover hashes the value a to a random number c using a random oracle.

3. The prover sends the value c to the verifier.

4. The verifier checks that $g^z = ay^e$ for some value of y. If this equation holds, the verifier is convinced that the prover knows the secret value u.

Applications of the Schnorr Protocol

The Schnorr protocol finds applications in various areas, including:

- Digital signatures: Efficient and secure generation of digital signatures.
- User authentication: Authentication of users in a privacy-preserving manner.
- Blind signatures: Generation of blind signatures, unlinkable to the signer.
- Batch verification: Efficient verification of a large number of signatures.

References

1. *Schnorr Non-interactive Zero-Knowledge Proof* by Nigel P. Smart

2. *Foundations of Cryptography: The Discrete Logarithm* by Feng Hao and Newcastle University (UK)