Abstract

Proofs-of-Work (PoW) are protocols that engage two parties, a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ in a manner that $\mathcal{V}$ can efficiently verify if $\mathcal{P}$ has computed a proof spending time T specified by $\mathcal{V}$. Such protocols have several important applications ranging from crypto-currencies to non-interactive time-stamping. However, a malicious prover $\mathcal{A}$ may bypass the specified delay T using massive parallelism to compute the proof. In order to counter such parallelism, a special class of PoW has been invented, namely proof-of-sequential-work (PoSW). These protocols mandate the prover $\mathcal{A}$ to spend time T even in the presence of polynomially many processors in T. This phenomenon is known as the sequentiality of the underlying computation. In general, PoSW may have multiple correct proofs but a subclass of it restricts the number of the correct proofs to be exponentially small in the security parameter. The PoSWs with such unique proofs are called verifiable delay functions (VDF).

The verification in each of the above-mentioned discipline of protocols is efficient. In particular, a proof computed in at least T-time must be verified in $\mathcal{O}(\log T)$-time. In this thesis, we address the question if the sequential effort required by the verifier can be optimized more? The answer is affirmative. We give concrete construction of a PoSW and two VDFs that require only a single sequential query to verify. In these proposed protocols, the number of sequential queries required in the verification is one and thus is independent of the delay parameter T and the security parameter λ.

In case of PoW (but not PoSW) the question of minimizing the verifier's sequential effort has been reasonably answered in the literature. There exists a one-way function based PoW that verifies its proof with a single sequential query. At the same time, we observe that the practical instantiation of this PoW deploys cryptographic hash functions demanding high power on the prover's side. In view of this, we optimize the energy requirement of the prover. Specifically, we design a PoW that uses less power required by the prover than that in the one-way function based PoW.