

Definition 1.1 (Dynamic weakly verifiable puzzle) A dynamic weakly verifiable puzzle (DWVP) is defined by a protocol between probabilistic algorithms (P, S) . The algorithm P is called a problem poser and S a problem solver. The problem poser P produces as output a circuit Γ_V and a circuit Γ_H . The circuit Γ_V takes as its input $q \in Q$ and an answer $r \in R$. An answer r is a correct solution for q if and only if the circuit Γ_V on input (q, r) evaluates to true. The circuit Γ_H on input q provides a hint $r \in R$ such that $\Gamma_V(q, r) = 1$. The solver S has oracle access to both circuits Γ_V and Γ_H . The calls to the circuit Γ_V are called verification queries. The calls to the circuit Γ_H are hint queries. The solver S asks at most h hint queries, v verification queries, and successfully solves a DWVP Π if and only if it makes a verification query (q, r) such that $\Gamma_V(q, r) = 1$, when it has not previously asked for a hint query on this q .

Suppose that $g : \{0, 1\}^k \rightarrow \{0, 1\}$ is a monotone function, and $(P^{(1)}, S^{(1)})$ is a dynamic weakly verifiable puzzle. Then $(P^{(g)}, S^{(g)})$ is a dynamic weakly verifiable puzzle $\Pi^{(g)}$, for which in the first phase the problem poser $P^{(g)}$ and solver $S^{(g)}$ sequentially create k instances of a puzzle $(P^{(1)}, S^{(1)})$. The problem poser $P^{(g)}$ produces as its output circuits $\Gamma_V^{(g)}$ and $\Gamma_H^{(g)}$. The hint queries for a puzzle $\Pi^{(g)}$ are answered by a circuit $\Gamma_H^{(g)}(q) = (\Gamma_H^{(1)}(q), \dots, \Gamma_H^{(k)}(q))$, and the verification queries by a circuit $\Gamma_V^{(g)}(q, r_1, \dots, r_k) = g(\Gamma_V^{(1)}(q, r_1), \dots, \Gamma_V^{(k)}(q, r_k))$.

Let $\text{hash} : Q \rightarrow \{0, 2(h+v)-1\}$ be a function and P_{hash} a set that contains elements $q \in Q$ for which $\text{hash}(q) = 0$. A *canonical success*, with respect to a set P_{hash} in a random experiment defined by the protocol between $P^{(g)}$ and $S^{(g)}$, is a situation when a first successful verification query made by $S^{(g)}$ is in P_{hash} , and all previous hint or verification queries are not in P_{hash} .

Theorem 1.2 (Security amplification for DWVP (non uniform version)). Let $g : \{0, 1\}^k \rightarrow \{0, 1\}$ be a monotone function, and $\text{hash} : Q \rightarrow \{0, 2(h+v)-1\}$ a function such that the probability of a canonical success, with respect to P_{hash} , is at least $\frac{\varepsilon}{8(v+h)}$. If there exists a circuit C that makes at most v verification queries, h hint queries, and succeeds with probability

$$\Pr[\Gamma_V^{(g)}(\langle P^{(g)}, C \rangle_C) = 1] \geq \Pr_{\mu \leftarrow \mu_\delta^k}[g(u) = 1] + \varepsilon, \quad (0.0.1)$$

where the probability is over randomness of $P^{(g)}$, then there exists a probabilistic algorithm $\text{Gen}(C, g, \varepsilon, \delta, n, \text{hash})$ which takes as input: a circuit C , a function g , a function hash , parameters ε, δ, n , and produces a circuit D of size at most $\text{size}(C)^{\frac{6k}{\varepsilon}} \log(\frac{6k}{\varepsilon})$ such that with high probability it satisfies

$$\Pr[\Gamma_V^{(1)}(\langle P^{(1)}, D \rangle_D) = 1] \geq \frac{1}{8(h+v)} \left(\delta + \frac{\varepsilon}{6k} \right) \quad (0.0.2)$$

where the probability is taken over random coins of P . Additionally, the circuit D and the algorithm Gen only require oracle access to functions g and hash .