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 problem solver S does not produce any output. 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 $g : \{0,1\}^k \to \{0,1\}$ is a monotone function, and $\left(P^{(1)}, S^{(1)}\right)$ 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 $\left(P^{(1)}, S^{(1)}\right)$. The problem poser $P^{(g)}$ produces as it output circuits $\Gamma_V^{(g)}$ and $\Gamma_H^{(g)}$. The hint queries for a puzzle $\Pi^{(g)}$ are answered by a circuit $\Gamma_H^{(g)}(q) = \left(\Gamma_H^{(1)}(q), \ldots, \Gamma_H^{(k)}\right)$, and the verification queries by a circuit $\Gamma_V^{(g)}(q, r_1, \ldots, r_k) = g\left(\Gamma_V^1(q, r_1), \ldots, \Gamma_V^k(q, r_k)\right)$.

Let $hash : Q \to \{0, 2(h+v) - 1\}$ be a function and P_{hash} a set that contains elements $q \in Q$ for which $hash(q) = 0$. 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 unifrom version)).*** *Let $g : \{0,1\}^k \to \{0,1\}$ be a monotone function, and $hash : Q \to \{0, 2(h+v) - 1\}$ a function such that the probability of a canonical success, with respect to P_{hash} in a random experiment defined by a protocol $(P^{(g)}, S^{(g)})$, 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*

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

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

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

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 $hash$, and the running time of the algorithm Gen is $poly\left(k, \frac{1}{\varepsilon}, n, v+h\right)$