Exercise 5.1 (Hardcore bit for the discrete logarithm). (6 points)

Let $G$ be a cyclic group of even order $d$ with a generator $g$, and let $\omega = g^{d/2}$. Furthermore suppose that an algorithm for computing square roots in $G$ is known. Let BitZero be a probabilistic algorithm that, given $g^i$, computes the least significant bit of $i$ in expected polynomial time.

The square root algorithm is given $g^{2i}$ with $0 \leq i < d/2$ and computes either the square root $g^i$ or the square root $\omega g^i$. Let Oracle be a probabilistic expected polynomial time algorithm that decides, which of the two square roots is $g^i$. [Note: This could be done by an oracle for the second least significant bit, $\text{bit}_1(i)$, of the discrete logarithm of $g^i$, where $0 \leq i < d$.]

(i) Formulate an algorithm for the discrete logarithm that uses at most polynomially many calls to Oracle and otherwise uses expected polynomial time. (Recall: The algorithm gets as input $g^i$ and should compute the discrete logarithm $d\log_g(g^i) = i$ with $0 \leq i < d$.)

(ii) What implications does this have on the security of ElGamal encryption scheme?