After a few runs of Shor's core subroutine with continued fractions, we have a c/d in lowest terms such that $|b/2^n - c/d| \leq 2^{-(n+1)}$. And we know that $c/d = \ell/p$, and we're trying to find p. If ℓ and p are coprime — and the chances are not bad — then $c = \ell$ and $d = p$ and we're done. But ℓ and p might not be coprime, in which case c and d are merely divisors of ℓ and p respectively. So check that $k^d \equiv 1 \pmod{m}$. If not, then run the core subroutine again, get a c'/d', and compute the least common multiple $\operatorname{lcm}(d, d')$. This LCM must divide p, and there's a good chance that it equals p. So check that $k^{\operatorname{lcm}(d,d')} \equiv 1 \pmod{m}$. If so, then $p = \operatorname{lcm}(d, d')$. If not, then start all over again.

There are many ways to tweak the details, but here is one complete, explicit rendering of the period-finding algorithm.

1. Input k and m.
2. Let n be the smallest integer such that $2^n \geq m^2$.
3. While p is unknown:
   (a) Set $d = m$ and $d' = m$.
   (b) While $d \geq m$:
      i. Run the core subroutine to obtain b.
      ii. Run continued fractions on $x_0 = b/2^n$, with larger and larger j, until you obtain c/d such that either $|b/2^n - c/d| \leq 2^{-(n+1)}$ or $d \geq m$.
   (c) If $k^d \equiv 1 \pmod{m}$, then output $p = d$.
   (d) While $d' \geq m$:
      i. Run the core subroutine to obtain b.
      ii. Run continued fractions on $x_0 = b/2^n$, with larger and larger j, until you obtain c'/d' such that either $|b/2^n - c'/d'| \leq 2^{-(n+1)}$ or $d' \geq m$.
   (e) If $k^{d'} \equiv 1 \pmod{m}$, then output $p = d'$.
   (f) Compute $\operatorname{lcm}(d, d') = d \cdot d' / \gcd(d, d')$.
   (g) If $k^{\operatorname{lcm}(d,d')} \equiv 1 \pmod{m}$, then output $p = \operatorname{lcm}(d, d')$.

Apparently the probabilities are such that very few iterations should be needed. For example, Nielsen and Chuang (2000, p. 231) argue that $p = \operatorname{lcm}(d, d')$ with probability at least 1/4.

Now suppose that $m = ab$, where a and b are distinct primes. The RSA cryptosystem is based on this kind of m, and knowing the factors of m breaks the cryptosystem. It turns out that period-finding and factoring are similar enough that the former gives a solution to the latter, as follows.

Pick a random k such that $2 \leq k < m$, and compute $\gcd(k, m)$. If the GCD is not 1, then congratulations; you just stumbled on a factor of m. So assume that k is coprime to m. Use the period-finding algorithm to find the smallest $p \geq 1$ such that $k^p \equiv 1 \pmod{m}$.

Now suppose that two pleasant things happen: p is even, and $k^{p/2} \not\equiv -1 \pmod{m}$. Because p is even, $p/2$ is an integer. We know that $k^{p/2} - 1$ is not divisible by m, because if it were then we'd have $k^{p/2} \equiv 1 \pmod{m}$ and p would not be the period. Meanwhile, to say that $k^{p/2} \not\equiv -1 \pmod{m}$ is to say that $k^{p/2} + 1$ is not divisible by m. So m does not divide $k^{p/2} - 1$ or $k^{p/2} + 1$, but m divides their product $(k^{p/2} - 1)(k^{p/2} + 1) = k^p - 1$. It follows that one of the primes a, b divides $k^{p/2} - 1$ and the other divides $k^{p/2} + 1$. So the GCD of m and either $k^{p/2} - 1$ or $k^{p/2} + 1$ produces either a or b.

If one (or both) of the pleasant things doesn't happen, then the number coming out of the GCD may not be a divisor of m. So proceed under the assumption that both pleasant things happen, but check your answer at the end, and re-run the algorithm if the answer is incorrect. Some basic number theory (Mermin, 2007, Appendix M) shows that the probability of both pleasant things happening is at least $1/2$. So we expect to try approximately two ks, and the probabilistic "worst case" isn't bad.