Problem 1-3. Hashing

In short, the attack is that the hash function of part (b) involves several steps that could lead to a collision whereas the hash function of part (a) only involves one, so we expect collisions to be statistically much more likely if the hash function is constructed as in part (b).

(a) The probability of a collision between $f\left(x_1\right)$ and $f\left(x_2\right)$ for some given $x_1 \neq x_2$ is simply $\frac{1}{2^n}$, the chance that the perfectly-random output of $f\left(x_2\right)$ happens to be the same as the (also perfectly-random) output of $f\left(x_1\right)$. The birthday paradox, then, finds that we expect a collision with roughly 50% probability in $\sqrt{2^n}$ attempts, so for $n = 160$, we expect a collision after on the order of 2^{80} attempts.

(b) Here, the probability of a collision between v_{i-1} and v_i for an x_1 and an x_2, for any i, is $\frac{1}{2^n}$, since (just like in the first part) we are using a random oracle g whose output is evenly distributed over 2^n possibilities. However, we run g once for each v_i from 1 to k. If the attacker holds all but the first block of x constant, and sets $x_1 = m_1||0^{b(k-1)}$ and $x_2 = m_2||0^{b(k-2)}$ for some $m_1 \neq m_2$, then there are k possibilities to see a hash collision in the final output. Specifically, a collision only does not occur in the output of h if one was avoided at every evaluation of g (every single $v_{i-1} \to v_i$ stage), so the final probability of a collision when calculating $v_k = h(M)$ is $1 - \left(1 - \frac{1}{2^n}\right)^k$. For $n = 160$ and $k = 1000$, we expect a collision after closer to 2^{75} attempts. If we see many collisions after 2^{80} attempts, we have reason to suspect that this is not a random oracle.