1. Assume the secret is \( X \). Assuming the two ZKPs are valid, the prover has to send values \( w_1 \) and \( w_2 \) satisfying:
\[
\begin{align*}
w_1 &= v + b_1 X \pmod{p-1} \\
w_2 &= v + b_2 X \pmod{p-1}
\end{align*}
\]
we can therefore recover \( X \) from the known quantities:
\[
X = (w_1 - w_2)(b_1 - b_2)^{-1} \pmod{p-1}
\]
assuming \( b_1 \neq b_2 \), which is almost surely true.

2. Let \( M \) be the original message, and the El-Gamal encrypted version of it be \((w = g^r, v = M h^r \pmod{p})\) for some random secret \( r \). Let \( S \) be the discrete log of \( h \) wrt \( g \) so that \( g^S = h \). Assume \( S \) has been Shamir secret-shared as \( n \) pieces \( S_1, \ldots, S_n \) where any \( t+1 \) pieces are enough to reconstruct \( S \). Assume wlog that the first \( t+1 \) users cooperate and send the server:
\[
y_i = w^{S_i}
\]
then this doesn’t expose information about \( S_i \) so long as discrete log is hard. Let \( L_i(0) \) be the Lagrange polynomial coefficients needed to reconstruct \( S \) from the \( S_i \), i.e. \( S = \sum_{i=1,\ldots,t+1} S_i L_i(0) \). The server computes:
\[
y = \prod_{i=1}^{t+1} y_i^{L_i(0)} = \prod_{i=1}^{t+1} w^{S_i L_i(0)} = w^S
\]
and can then recover the message \( M \) as \( vy^{-1} \pmod{p} \).