Problem 1.

Part A Construction:

1. The prover $\mathcal{P}$ samples $r_1, \ldots, r_n$ and sends $\alpha_i = \prod_{j=1}^n g_{ij}^{r_j}$ for $i = 1, \ldots, m$.
2. $\mathcal{V}$ chooses a challenge $c \in \mathbb{Z}_p$ and sends it to $\mathcal{P}$.
3. $\mathcal{P}$ calculates $s_j = cx_j + r_j$ and sends $s_1, \ldots, s_n$ to $\mathcal{V}$.
4. $\mathcal{V}$ accepts if $\prod_{j=1}^n g_{ij}^{s_j} = u_i^c \alpha_i$ for $i = 1, \ldots, m$.

Completeness: $\mathcal{V}$ always outputs accepts since

$$\prod_{j=1}^n g_{ij}^{s_j} = \prod_{j=1}^n g_{ij}^{cx_j + r_j} = \left(\prod_{j=1}^n g_{ij}^{x_j}\right)^c \prod_{j=1}^n g_{ij}^{r_j} = u_i^c \alpha_i.$$

Soundness: When $(\{g_{ij}\}, \{u_i\}) \notin \mathcal{L}$. For any $\alpha_1, \ldots, \alpha_m$, it is impossible to have $c \neq c'$ so that there exist s_j, s'_j with

$$\prod_{j=1}^n g_{ij}^{s_j} = u_i^c \alpha_i \text{ and } \prod_{j=1}^n g_{ij}^{s'_j} = u_i^{c'} \alpha_i$$

Else we can take $x_j = (s_j - s'_j)(c - c')^{-1}$, then we have $(\{g_{ij}\}, \{u_i\}) \in \mathcal{L}$. So the Soundness error is less than $1/p$.

Zero knowledge: Simulator $\mathcal{S}$ samples $c \in \mathbb{Z}_p$ and $s_j \in \mathbb{Z}_p$ for $j = 1, \ldots, n$, computes $\alpha_i = \prod_{j=1}^n g_{ij}^{s_j} u_i^{-c}$, and outputs $(\{\alpha_i\}, c, \{s_j\})$ as an perfect simulated transcript.

Part B The extractor works as follows:

1. Get the first message from $\mathcal{P}$, $\alpha_1, \ldots, \alpha_m$.
2. Randomly choose two distinct challenges c, c'.
3. Get the answers $\{s_j\}, \{s'_j\}$ by rewinding the prover.
4. Calculate $x_j = (s_j - s'_j)/(c - c')$ in $\mathbb{Z}_p$ for $j = 1, \ldots, n$.

Part C The prover knows x, y satisfying

$$a = g^x, \quad b = g^y, \quad c = b^x, \quad c = a^y.$$

This is a linear formula which has a ZK proof of knowledge system, as shown in part A and part B.

Part D Since $v_1^y = g^{\beta_1 y}$ and $v_3 = g^{\beta_3}$, we have $v_1^y v_3^{-1} = g^{\beta_1 y - \beta_3}$. Since $e_1^y = u^{\beta_1 y} g^{xy}$ and $e_3 = u^{\beta_3} g^{xy}$, we have $e_1^y e_3^{-1} = u^{\beta_1 y - \beta_3}$. Take $w = \beta_1 y - \beta_3$.

$$\begin{cases} v_1 = g^{\beta_1} \\ v_1^y v_3^{-1} = g^{\beta_1 y - \beta_3} \\ v_2 = g^{\beta_2} \\ e_1 = u^{\beta_1} g^x \\ e_1^y e_3^{-1} = u^{\beta_1 y - \beta_3} \\ e_2 = u^{\beta_2} g^y \end{cases} \iff \begin{cases} v_1 = g^{\beta_1} \\ v_1^y v_3^{-1} = g^w \\ v_2 = g^{\beta_2} \\ e_1 = u^{\beta_1} g^x \\ e_1^y u^{-w} = e_3 \\ e_2 = u^{\beta_2} g^y \end{cases}$$

The above equation is linear, so it can be proved use the construction in Part A. Here we take $(x, y, w, \beta_1, \beta_2)$ as the witness. The existence of $(\beta_1, \beta_2, x, y, w)$ is equivalent to the existence of $(\beta_1, \beta_2, \beta_3, x, y)$, because we can solve β_3 by $\beta_3 = \beta_1 y - w$

Part E Suppose $v = g^\beta, v' = g^{\beta'}$. Take $y_d = \lambda_d$, $y_{i-1} = \lambda_{i-1} + x y_i$ for $i = d, d-1, \ldots, 1$. Then $y_0 = f(x)$. The existence of $(\beta, \beta', y_0, \ldots, y_d)$ is equivalent to the existence of (β, β', x). We prove the existence of $(\beta, x, y_d, \ldots, y_0)$ so that:

$$\begin{cases} v = g^\beta \\ e = u^\beta g^x \\ u' = g^{\beta'} \\ e' = u^{\beta'} g^{y_0} \\ g^{y_d} = g^{\lambda_d} \\ g^{y_{i-1}} = g^{\lambda_{i-1}} g^{x y_i} \text{ for } i = 1, \ldots, d \end{cases} \iff \begin{cases} v = g^\beta \\ e = u^\beta g^x \\ u' = g^{\beta'} \\ e' = u^{\beta'} g^{y_0} \\ g^{y_d} = g^{\lambda_d} \\ g^{y_{i-1}} u^{\beta y_i} = g^{\lambda_{i-1}} e^{y_i} \text{ for } i = 1, \ldots, d \end{cases}$$

This step is by multiply $u^{\beta y_i}$ on both sides of the last equation $g^{y_{i-1}} = g^{\lambda_{i-1}} g^{x y_i}$ and plug in the second equation $e = u^\beta g^x$. Therefore, the equation system is transformed to a linear system and can be proved using the construction in Part A.

Part F Notice that $b \in \{0, 1\} \iff b^2 = b$. So we set $f(x) = x^2$ and prove that (v, e) is also an encryption of b^2 using the construction in Part E.

Problem 2.

In the opening phase, the sender $\mathcal{S}$ sends (m, r). If $c = h^m g^r$, the receiver accepts.

Perfect hiding: for any m, $h^m g^r$ is uniformly distributed in $\mathbb{G}$.

Computational binding: if the sender can output accepted (m', r') with $m \neq m'$, then we can solve discrete log of h based on the sender: $h^{m'} g^{r'} = h^m g^r$ implies that the discrete log $\log_g(h) = \frac{r-r'}{m'-m}$. Since the hardness of DDH problem in $\mathbb{G}$ implies the hardness of discrete log problem in $\mathbb{G}$, the protocol should be computational binding.