Problem 1.

Part A. WLOG assume that $l = 1$, otherwise we just send each bit separately. For each $i \in \{1, \dots, t\}$, the receiver chooses $b = 1$ if $i = x$ and $b = 0$ otherwise; the sender chooses $\hat{m}_0 = 0$ and $\hat{m}_1 = m_i$. Then the receiver uses the 1-out-of-2 OT to get $v_i = \hat{m}_b$. The result is $v_1 \vee \cdots \vee v_t$.

To simulate the view of sender, we just call $\mathsf{OT.Sim}_S\left((0, m_i), \perp\right)$ for $i \in \{1, \dots, t\}$. To simulate the view of receiver, we just call $\mathsf{OT.Sim}_T(0, 0)$ for $i \neq x$ and $\mathsf{OT.Sim}_T\left(1, m_x\right)$ for $i = x$. Therefore the scheme is secure.

Part B. The protocol goes like this:

- The sender prepares t random messages $r_1, \cdots, r_t$ uniformly sampled from $\{0,1\}^\ell$.
- The given 1-out-of-2 OT protocol is used for t rounds. In round i, the sender prepares the following two inputs for the OT protocol:

$$\left(r_i, \bigoplus_{j<i} r_j \oplus m_i\right)$$

- If the receiver wants to learn m_x, he or she requires the former message in the first $x-1$ rounds, and the later message in round x, which means he or she can learn

$$r_1, \cdots, r_{x-1}, \bigoplus_{j<x} r_j \oplus m_x$$

  helping him or her reveal m_x.

The correctness for this protocol is obvious. The view of the sender can be simulated by t $\mathsf{OT.Sim}_S$, which runs $\mathsf{OT.Sim}_S$ on t pairs of inputs prepared by the sender independently.

The view of the receiver can also be simulated by t $\mathsf{OT.Sim}_R$. In the first $x-1$ rounds, the (semi-honest) receiver can only require the former message, and t OT. Sim $_R$ simply samples $r \leftarrow \$$ and returns $\mathsf{OT.Sim}_R(r, 0)$ to the receiver. Of course it should remember all r-s. When round x comes, the receiver requires the later message, the t $\mathsf{OT.Sim}_R$ will return $\mathsf{OT.Sim}_R\left(m_x \oplus R, 1\right)$, where R denotes the XOR sum of all r-s. In the remaining rounds, it simply returns OT.$\mathsf{Sim}_R(\$, b)$ for the require bit b from the receiver. Thus, the protocol is secure against semi-honest sender and semi-honest receiver.

Problem 2.

Part A, Solution 1. Define f as $f((k_1, k_2), x) = f_2(k_2, f_1(k_1, x))$, where $f_1 : \{0,1\}^{\ell_{\text{key1}}} \times \{0,1\}^* \to \{0,1\}^n$ is a UHF and $f_2 : \{0,1\}^{\ell_{\text{key2}}} \times \{0,1\}^n \to \{0,1\}^{2n}$ is a PRF. (In short, we want $f : \{0,1\}^{\ell_{\text{key}}} \times \{0,1\}^* \to \{0,1\}^{2n}$ be a PRF for arbitrary length input.)

The output of f is interpreted as two field elements in $\mathbb{F}_{2^n}$.

> $\mathsf{Gen}(1^n, i)$ samples a random PRF key for f.
>
> $\mathsf{MAC}((k_1, \ldots, k_n), m)$ computes $(x_i, y_i) = f(k_i, m)$ for each $i \in [n]$, finds the degree-$(n-1)$ polynomial t such that $t(x_i) = y_i$ for all $i \in [n]$.
>
> $\mathsf{Verify}(i, k, m, t)$ computes $(x, y) = f(k, m)$, and accepts the tag if and only if $t(x) = y$.

Part A, Solution 2. Define f as $f((k_1, k_2), x) = f_2(k_2, f_1(k_1, x))$, where $f_1 : \{0,1\}^{\ell_{\text{key1}}} \times \{0,1\}^* \to \{0,1\}^n$ is a UHF and $f_2 : \{0,1\}^{\ell_{\text{key2}}} \times \{0,1\}^n \to \{0,1\}^{n(n+1)}$ is a PRF. (In short, we want $f : \{0,1\}^{\ell_{\text{key}}} \times \{0,1\}^* \to \{0,1\}^{n(n+1)}$ be a PRF for arbitrary length input.)

The output of f is interpreted as a dimension-$(n+1)$ vector in $(\mathbb{F}_{2^n})^{n+1}$.

> $\mathsf{Gen}(1^n, i)$ samples a random PRF key for f.
>
> $\mathsf{MAC}((k_1, \ldots, k_n), m)$ computes $v_i = f(k_i, m)$ for each $i \in [n]$, a vector t which is orthogonal to all of $v_1, \ldots, v_n$.
>
> $\mathsf{Verify}(i, k, m, t)$ computes $v_i = f(k, m)$, and accepts the tag if and only if v_i is orthogonal to t.

Part B. For one-time security, f_2 can be replaced by a 2-universal hash function.

Problem 3.

Use the following protocol to compute a function f

- Let $(\mathsf{Gen}, \mathsf{MAC}, \mathsf{Verify})$ be a one-time MAC scheme. The i-th party samples key $k_i \leftarrow \mathsf{Gen}(1^\lambda)$.
- Use a PKO secure protocol to compute f', which is defined as

  $$f'((x_1, k_1), \ldots, (x_n, k_n)) = ((y_1, \mathsf{MAC}(k_1, y_1), \ldots, (y_n, \mathsf{MAC}(k_n, y_n)),$$

  where $(y_1, \ldots, y_n) = f(x_1, \ldots, x_n)$. By assumption, such PKO secure protocol exists, denote the protocol by Π'. All honest parties learn $(\hat{y}, \hat{t})$ from Π'.
- The i-th party outputs $\hat{y}$ if $\mathsf{Verify}(k_i, \hat{y}, \hat{t})$ accepts. Otherwise, the i-th party aborts (i.e., outputs $\perp$).