Recall the definitions of KPA and CPA secure encryption:

**Definition 1** A pair of algorithms Enc, Dec, and ensembles $K = \{K_n\}_{n \in \mathbb{N}}, M = \{M_n\}_{n \in \mathbb{N}}$ are a $t$-KPA secure encryption scheme w.r.t. leakage function $l()$ if:

**Correctness:** For any $n \in \mathbb{N}$ and for any $k \in K_n$ and $m \in M_n$ we have $\text{Prob}(m = \text{Dec}(k, \text{Enc}(k, m))) = 1$.

**Secrecy:** For any two ensembles of message vectors $m_0 = \{m_{0,n}\}_{n \in \mathbb{N}}$, $m_1 = \{m_{1,n}\}_{n \in \mathbb{N}}$, where $m_{0,n} = m_{0,n,1},...,m_{0,n}$, $m_{1,n} = m_{1,n,1},...,m_{1,n}m_{1,n}$ and each $m_{i,j}^k \in M_n$ and $l(m_{0,n}) = l(m_{1,n})$, we have:

\[ \{[k \leftarrow K_n : \text{Enc}(k, m_{0,n}),...,\text{Enc}(k, m_{1,n})] \}_{n \in \mathbb{N}} \approx \{[k \leftarrow K_n : \text{Enc}(k, m_{1,n}),...,\text{Enc}(k, m_{1,n})] \}_{n \in \mathbb{N}}. \]

The scheme is KPA secure (w.r.t. $l()$) if it is $t$-KPA secure for any polynomial $t(n)$.

**Definition 2** A pair of algorithms Enc, Dec, and ensembles $K = \{K_n\}_{n \in \mathbb{N}}, M = \{M_n\}_{n \in \mathbb{N}}$ are a $t$-CPA secure encryption scheme w.r.t. leakage function $l()$ if:

**Correctness:** For any $n \in \mathbb{N}$ and for any $k \in K_n$ and $m \in M_n$ we have $\text{Prob}(m = \text{Dec}(k, \text{Enc}(k, m))) = 1$.

**Secrecy:** For any polytime adversary $\{A_n\}$ there exists a negligible function $\nu()$ such that $A_n$ wins the $t$-CPA game w.r.t. leakage $l()$ with probability at most $1/2 + \nu(n)$. The $t$-CPA game proceeds as follows:

1. A key $k$ is chosen uniformly from $\{0,1\}^n$. A random bit $b$ is chosen.
2. Repeat for $t$ times:
   (a) $A_n$ outputs $m_0, m_1$ s.t. $l(m_0) = l(m_1)$
   (b) $A_n$ obtains $\text{Enc}(k, m_b)$
   $A_n$ outputs a bit $b'$, and wins if $b = b'$.

The scheme is CPA secure (w.r.t. $l()$) if it is $t$-CPA secure w.r.t. $l()$ for any polynomial $t(n)$.

**Question 1 (60%)**: Prove or disprove: Any encryption scheme that’s KPA secure with respect to leakage function $l()$ is also CPA secure with respect to $l()$. Note: Proving the statement amounts to coming up constructing an adversary that breaks the KPA security of a scheme, given an adversary that breaks the CPA security of the scheme. Disproving the statement amounts to demonstrating an encryption scheme that is KPA secure but not CPA secure. (You can construct the scheme as strangely as you wish, as long as it is KPA but not CPA.)

Bonus (30 %) : Show a stateless encryption scheme that is 2-CPA secure but not CPA secure. (Can you show a scheme that is 2-CPA secure but not 3-CPA secure?)

**Question 2 (60%)**:

A family of functions $H = \{h_k : \{0,1\}^n \rightarrow \{0,1\}^m\}_{k \in \{0,1\}^n}$ is called a universal hash family if for any $\alpha, \beta \in \{0,1\}^n, \alpha \neq \beta$, $\text{Prob}_{k \leftarrow \{0,1\}^n}[h_k(\alpha) = h_k(\beta)] = 1/2^m$. (That is, the probability that two fixed points in the domain collide under $h_k$ is exactly the same as if $h_k$ were a truly random function from $\{0,1\}^n$ to $\{0,1\}^m$.) There are many combinatorial constructions of universal hash functions families with relatively short keys.

1. **[20 points]** Show that the family $\{h_A(x) = Ax\}_{A \in \mathbb{F}_2^{nxm}, b \in \{0,1\}^m}$, where $A_{nxm}$ is the set of $n$ by $m$ binary matrices, and the arithmetic is done in $\mathbb{F}_2$, is universal hash. (Arithmetic in $\mathbb{F}_2$ means reducing each resulting integer modulo 2.)
2. [40 points] An ensemble $H = \{H_n\}_{n \in \mathbb{N}}$ of families of functions is universal with range $m(n)$ if for each $n \in \mathbb{N}$ the family $H_n$ is a universal hash family of functions from $\{0, 1\}^n$ to $\{0, 1\}^{m(n)}$.

For any constant $c > 0$, show how to modify the GGM construction of pseudorandom function families so that each evaluation of the function on inputs in $\{0, 1\}^n$ will involve only $cn$ applications of the underlying length-doubling pseudorandom generator. (Here $n$ is taken to be the security parameter.) Hint: Use universal hash family ensembles.