Answer to Question 1.
Suppose that Cryptosystem VII does not satisfy Privacy. We will show that $F$ is not strongly pseudo-random.

Let $\{C_n\}$ be an adversary that breaks the Privacy of Cryptosystem VII. Define $q_C(n)$ as in the definition of Privacy. Then $q_C(n) > \frac{1}{2} + \frac{1}{n^c}$ for some $c$ and infinitely many $n$. Fix $n$ such that $q_C(n) > \frac{1}{2} + \frac{1}{n^c}$. Let $i_n = i$ be the index of the piece that $C_n$ is trying to guess the decryption of.

Consider an alternative experiment where instead of using $F_k$ for random $k$ to form the encryptions $e_j = F_k(jm_j)$, we use a randomly chosen permutation $f$, so that $e_j = f(jm_j)$; similarly, $B$ uses $f^{-1}$ to decrypt. In this case, when seeing encryptions of pieces, the only information the adversary sees about $b$ is $f(\tilde{im}^b)$, where none of the other encryptions are of the form $f(\tilde{im})$ for $m \in \{0,1\}^n$. But this means that the adversary does not get any information at all about $b$ when he sees the encryptions of pieces (since for randomly chosen $f$, we have that $f(\tilde{im}^b)$ and $f(\tilde{im}^1)$ are distributed identically). Also, note that the adversary does not gain any information about $b$ when he sees if, and for which piece, $B$ outputs FAIL. This is because $B$ outputs FAIL whenever he decrypts a piece $e_j'$ such that the leftmost $n$ bits of $f^{-1}(e_j')$ are not $\tilde{j}$. So seeing that $B$ outputs FAIL for a piece $e_j'$ only gives the adversary information about the leftmost $n$ bits of $f^{-1}(e_j')$. But the adversary is trying to gain information about the rightmost $n$ bits of $f^{-1}(f(\tilde{im}^b))$; he already knows that the first $n$ bits of $f^{-1}(f(\tilde{im}^b))$ are $\tilde{i}$. So the adversary gets no information at all about $b$, and hence the probability that he guesses $b$ correctly is $1/2$.

We will describe a strong adversary $D_n$ for distinguishing $F$. $D_n$ will be given a black box for a permutation $f : \{0,1\}^{2n} \rightarrow \{0,1\}^{2n}$ as well as a black box for $f^{-1}$.

$D_n$ will simulate $A$, $B$, and $C_n$. Whenever $C_n$ wants an encryption of a $j$-th piece $m_j$, $D_n$ will create $e_j = f(jm_j)$. Also, $D_n$ will simulate the person choosing the random bit $b$. Once $C_n$ outputs a string $\alpha$ for a decryptor, $D_n$ will (using $f^{-1}$) simulate $B$ running on $\alpha$ in order to determine if, and for which piece, $B$ outputs FAIL; this result will be given to $C_n$. $C_n$ will then output a bit $b'$. $D_n$ will accept if $b' = b$, and $D_n$ will reject otherwise.

Let $p_D(n)$ be the probability that if $k$ is randomly chosen from $\{0,1\}^n$ and $D_n$ is given black boxes for $F_k$ and $F_k^{-1}$, then $D_n$ accepts. Then, $p_D(n) = q_C(n) > \frac{1}{2} + \frac{1}{n^c}$.

Let $r_D(n)$ be the probability that if $f$ is a randomly chosen permutation mapping $\{0,1\}^n$ to $\{0,1\}^n$ and $D_n$ is given black boxes for $f$ and $f^{-1}$, then $D_n$ accepts. Then, $r_D(n) = \frac{1}{2}$.

Answer to Question 2.

a. Let $D$ be the distinguisher that given $y \in \{0,1\}^{n+1}$, accepts if and only if $t(G(A(t(y)))) = t(y)$.

We first calculate the probability that $D$ accepts a pseudo-randomly generated $n + 1$-bit string. That is, we want to calculate the probability that $t(G(A(t(G(x))))) = t(G(x))$ when $x$ is randomly chosen from $\{0,1\}^n$. But this is just the probability that $f(A(f(x))) = f(x)$, which is $q(n)$.

We next calculate the probability that $D$ accepts a randomly generated $n + 1$-bit string. That is, we want to calculate the probability that $t(G(A(t(y)))) = t(y)$ when $y$ is randomly chosen from
\{0,1\}^{n+1}. Since \(y\) randomly chosen from \(\{0,1\}^{n+1}\) implies \(t(y)\) is randomly chosen from \(\{0,1\}^{n-1}\), this is equal to the probability that \(t(G(A(z))) = z\) when \(z\) is randomly chosen from \(\{0,1\}^{n-1}\), which is \(C(n)/2^{n-1}\).

For infinitely many \(n\), the difference between these two probabilities is \(> 1/(2n^d)\).

b. Let \(X = \{x \in \{0,1\}^n \mid G(A(t(G(x)))) = G(x)\}\).

Let \(Z = \{z \in \{0,1\}^{n-1} \mid t(G(A(z))) = z\}\).

To show \(|X| \geq |Z|\), it is sufficient to give a 1-1 mapping from \(Z\) to \(X\). We will show that \(A\) is such a mapping.

We first show that \(A\) maps \(Z\) to \(X\). Let \(z \in Z\). Then \(t(G(A(z))) = z\). But then \(G(A(t(G(A(z)))) = G(A(z))\). That is, \(A(z) \in X\).

To see that this mapping is 1-1, note that if \(z_1, z_2 \in Z\) such that \(A(z_1) = A(z_2)\), then \(z_1 = t(G(A(z_1))) = t(G(A(z_2))) = z_2\).

c. Let \(Y = \{y \in \{0,1\}^{n+1} \mid G(A(t(y))) = y\}\).

Let \(Z = \{z \in \{0,1\}^{n-1} \mid t(G(A(z))) = z\}\).

To show \(|Y| \leq |Z|\), it is sufficient to give a 1-1 mapping from \(Y\) to \(Z\). We will show that \(t\) is such a mapping.

We first show that \(t\) maps \(Y\) to \(Z\). Let \(y \in Y\). Then \(G(A(t(y))) = y\). But then \(t(G(A(t(y)))) = t(y)\). That is, \(t(y) \in Z\).

To see that this mapping is 1-1, note that if \(y_1, y_2 \in Y\) such that \(t(y_1) = t(y_2)\), then \(y_1 = G(A(t(y_1))) = G(A(t(y_2))) = y_2\).

d. Let \(D\) be the distinguisher that given \(y \in \{0,1\}^{n+1}\), accepts if and only if \(G(A(t(y))) = y\).

The probability that \(D\) accepts a pseudo-randomly generated \(n+1\)-bit string is \(|\{x \in \{0,1\}^n \mid G(A(t(G(x)))) = G(x)\}|/2^n \geq C(n)/2^n\) (by Part b).

The probability that \(D\) accepts a randomly generated \(n+1\)-bit string is \(|\{y \in \{0,1\}^{n+1} \mid G(A(t(y))) = y\}|/2^{n+1} \leq C(n)/2^{n+1}\) (by Part c).

For infinitely many \(n\), the difference between these two probabilities is \(\geq C(n)/2^{n+1} = C(n)/(4 \cdot 2^{n-1}) > 1/(8n^d)\).

Answer to Question 3.

a. Fix \(k_1, k_2 \in \{0,1\}^n\).

Let \(L_1R_1 = F'_{k_1k_2}(\bar{0}\bar{0})\). Then \(L_1 = F_{k_1}(\bar{0})\). Now consider \(L_2R_2 = F'_{k_1k_2}(L_1\bar{0})\). We have \(L_2 = L_1 \oplus F_{k_1}(\bar{0}) = \bar{0}\).

Define the following distinguisher \(D\) for \(F'\): Given \(f : \{0,1\}^{2n} \to \{0,1\}^{2n}\), let \(L_1R_1 = f(\bar{0}\bar{0})\) and accept if the first \(n\) bits of \(f(L_1\bar{0})\) are all 0.

Then \(p_D(n) = 1\).

Now consider \(r_D(n)\). Note that if \(L_1 = \bar{0}\) then \(D\) will always accept; for randomly chosen \(f\), we have \(L_1 = \bar{0}\) with probability \(1/2^n\). If \(L_1 \neq \bar{0}\), the probability that the first \(n\) bits of \(f(L_1\bar{0})\) are all 0 when \(f\) is randomly chosen is \(1/2^n\). So \(r_D(n) \leq 2/2^n\).

b. We are given oracles for two permutations \(f, f^{-1} : \{0,1\}^n \to \{0,1\}^n\); either \(f\) has been chosen randomly, or \(f = F''_{k_1k_2k_3}\) for randomly chosen \(k_1, k_2, k_3\).

We are looking for evidence that \(f = F''_{k_1k_2k_3}\).
As in the hint, we want to find two distinct $n$-bit strings $Y_1$ and $Y_2$ such that we also know the value of $F_{k_1}(Y_1) \oplus F_{k_1}(Y_2)$.

So choose three $n$-bit strings $L, R_1, R_2$ such that $R_1 \neq R_2$, and let

$$ (W_1 Y_1) = f^{-1}(LR_1) $$
$$ (W_2 Y_2) = f^{-1}(LR_2). $$

Then $F_{k_1}(Y_1) \oplus F_{k_1}(Y_2) = W_1 \oplus W_2 \oplus R_1 \oplus R_2$. More conveniently, we can write

$$ (W_2 \oplus R_1 \oplus R_2) \oplus F_{k_1}(Y_2) = W_1 \oplus F_{k_1}(Y_1). $$

So let

$$ (UV) = f((W_2 \oplus R_1 \oplus R_2) Y_2); $$

recall that we know that (LR_1) = f(W_1 Y_1).

So accept if $U \oplus Y_2 = L \oplus Y_1$, and reject otherwise.

This algorithm accepts a pseudo-randomly generated $f$ with probability 1.

Assume now that $f$ is a randomly chosen permutation; we want to prove that the probability that our algorithm accepts is very small. We can view the experiment as follows: whenever we make a query whose result is not determined by previous queries, then we choose a random answer from amongst those that have not been ruled out.

To start with, our query $f^{-1}(LR_1)$ is answered by a random $(W_1 Y_1)$. Since $(LR_2) \neq (L, R_1)$, the query $f^{-1}(LR_2)$ is answered by a random $(W_2, Y_2)$ subject only to the constraint that $(W_2, Y_2) \neq (W_1, Y_1)$. So almost certainly $Y_1 \neq Y_2$. We now make the query $f((W_2 \oplus R_1 \oplus R_2) Y_2)$. We have already specified the value of $f$ on two inputs: $(W_1 Y_1)$ and $(W_2 Y_2)$. But since $Y_1 \neq Y_2$ and $R_1 \oplus R_2 \neq 0$, it must be the case that $((W_2 \oplus R_1 \oplus R_2) Y_2)$ is not equal to either of these queries. So $f((W_2 \oplus R_1 \oplus R_2) Y_2)$ is answered with a random $(UV)$, subject only to the constraint that $(UV)$ is equal to neither $(LR_1)$ nor $(LR_2)$. So almost certainly, $U$ will not be equal to $L \oplus Y_1 \oplus Y_2$ and the algorithm will reject.

**Answer to Question 4.** Assume that a secure public-key signature scheme exists. Let $GEN$ be the key generating function. $GEN$ takes a security parameter $1^n$ and a string $r$ of (say) $n^c$ random bits, and outputs two keys each of length (say) $n^d$: $pub$ and $pri$.

Now define the function $f$ as follows: on input $r$ of length $n^c$, $f(r) = pub$ where $pub$ is the first part of the output of $GEN(1^n, r)$. We claim $f$ is one-way.

Assume we had an adversary $A$ for breaking the one-wayness of $f$. We could then break the signature scheme as follows: Given $pub$ of length $n^d$, compute $r = A(pub)$, and compute $G(1^n, r) = pub', pri$. If the adversary was successful in getting $r$ such that $f(r) = pub$, then $pub' = pub$; since the scheme satisfies correctness, $pri$ can now be used to sign any message one wants.