4.7

b) Consider an arbitrary adversary A that has oracle access to Mac. A requests the tags for $m_a = 0^n, m_b = 0^{n/2}1^{n/2}, m_c = 1^n$:

$t_a = F_k(<1>||0^{\frac{n}{2}}) \oplus F_k(<2>||0^{n/2})$

$t_b = F_k(<1>||0^{\frac{n}{2}}) \oplus F_k(<2>||1^{n/2})$

$t_c = F_k(<1>||1^{\frac{n}{2}}) \oplus F_k(<2>||1^{n/2})$

Now by XORing those tags together:

$t_a \oplus t_b \oplus t_c = F_k(<2>||0^{\frac{n}{2}}) \oplus F_k(<1>\Big|\Big|1^{\frac{n}{2}}\Big) = F_k(<1>||1^{\frac{n}{2}}) \oplus F_k(<2>||0^{\frac{n}{2}})$

This corresponds to Mac($1^{n/2}0^{n/2}$).

A is able to ouput a message $m = 1^{n/2}0^{n/2}$ and a tag $t = t_a \oplus t_b \oplus t_c$ that was not previously requested, such that Vrfy(m,t)=1

c) Consider an arbitrary adversary A that has oracle access to Mac. Consider an arbitrary message $m \in \{0,1\}^n, m = m_1||m_2$ with $|m_1| = |m_2| = n/2$.

Notice that if $r = <1>||m_1$ then $t = F_k(<1>||m_1) \oplus F_k(<1>||m_1)$, because $|m_1| = n/2$
$= 0^n$

Now in this scheme r is chosen uniformly, so if $r = <1>||m_1$ is valid.

A outputs $m = m_1$ *and tag* $= < \ <1>||m_1, 0^n \ >)$ and succeeds all the time : Vrfy(m,t)=1

4.13

a) An adversary A requests the CBC-MAC oracle for the tag on $m1 = m1_1 m1_2$ with |m1|=2n, and obtains tag $t1_2$: $t1_1 = F_k(0^n \oplus m1_1) = F_k(m1_1)$ *then* $t1_2 = F_k(F_k(m1_1) \oplus m1_2)$

Now for some $m2 = m2_1 m2_2$ *with* $|m2| = 2n$, A requests the tag on $t1_2 \oplus m2_1$ and obtains tag $t2_2$:

$t2_1 = F_k(0^n \oplus t1_2 \oplus m2_1) = F_k(t1_2 \oplus m2_1)$

$t2_2 = F_k(F_k(t1_2 \oplus m2_1) \oplus m2_2)$

Notice that the tag for $m1||m2$ is t_4 :

$t_1 = F_k(0^n \oplus m1_1)$, $\quad t_2 = F_k(F_k(m1_1) \oplus m1_2) = t1_2$,

$t_3 = F_k(t1_2 \oplus m2_1)$, $\quad t_4 = F_k(F_k(t1_2 \oplus m2_1) \oplus m2_2) = t2_2$

Therefore A is able to ouput a message $m = m1||m2$ and a tag $t = t2_2$ that was not previously requested, such that Vrfy(m,t)=1

b) An adversary A requests the CBC-MAC oracle for the tag on $m1 = m1_1 m1_2$ with |m1|=2n, and obtains tag $t1_2$: $t1_1 = F_k(0^n \oplus m1_1) = F_k(m1_1)$
then $t1_2 = F_k(F_k(m1_1) \oplus m1_2)$

Now for some $m2$ *with* $|m2| = n$, A requests the tag on $t1_2 \oplus m2$ and obtains tag $t2$:

$t2 = F_k(0^n \oplus t1_2 \oplus m2) = F_k(t1_2 \oplus m2)$

Notice that the tag for $m1||m2$ is $t_3 = t2$:

$t_1 = F_k(0^n \oplus m1_1)$, $\quad t_2 = F_k(F_k(m1_1) \oplus m1_2) = t1_2$,

$t_3 = F_k(t1_2 \oplus m2) = t2$

Therefore A is able to ouput a message $m = m1\big|\big|m2$ *with* $\big|m\big| = 3n$ and a tag $t = t2$ that was not previously requested, such that Vrfy(m,t)=1

4.14

An adversary A requests the CBC-MAC oracle for the tag on $m = 0^n$, and obtains <t0, t1> where $t1 = F_k(t0 \oplus 0^n) = F_k(t0)$.

Now for some m'≠m with |m'|=n, A outputs (m', <m'$\oplus t0$,t1>). t0 is a random block of length n, so m'$\oplus t0$ is valid.

Moreover will obtain Vrfy(m', <m'$\oplus t0$,t1>)=1 since :

$F_k(m' \oplus t0 \oplus m') = F_k(0^n \oplus t0) = t1$, and m' was not requested before.

10.3

If an adversary A can intercept $h_A = g^x$ sent from Alice, then A can choose its own $x' \in \mathbb{Z}q$ (since G,q,g are standardized and known) and send to Bob $h'_A = g^{x'}$. Bob cannot notice that this is not from Alice since he doesn't know x. Now when Bob sends $h_B = g^y$, A can intercept it and choose its own $y' \in \mathbb{Z}q$. A sends to Alice $h'_B = g^{y'}$. Alice cannot notice that this is not from Bob since she doesn't know y.

A shares the key $k_A = {h'_B}^x = g^{y'x}$ with Alice.

A shares the key $k_B = {h'_A}^y = g^{x'y}$ with Bob.

10.4

Alice outputs k.

Bob outputs $w \oplus t = u \oplus r \oplus t = s \oplus t \oplus r \oplus t = k \oplus r \oplus r = k$

Therefore both parties have the same key.

This protocol is insecure. Consider a man-in-the-middle attack where an adversary A can intercept and modify messages.

A intercepts $s = k \oplus r$ sent by Alice, and sends to Bob $s' = k' \oplus r'$.

Bob sends $u = s' \oplus t$. A intercepts u and sends to Alice $u' = s \oplus t'$.

Alice send w= $u' \oplus r$. A intercepts w and sends to Bob $w' = u \oplus r'$.

Bob outputs $w' \oplus t = u \oplus r' \oplus t = s' \oplus t \oplus r' \oplus t = k' \oplus r' \oplus t \oplus r' \oplus t = k' \oplus t \oplus t = k'$ which is known by A.

Alice outputs k. A can compute k with $w \oplus t' = u' \oplus r \oplus t = s \oplus t' \oplus r \oplus t' = s \oplus r = k \oplus r \oplus r = k$.

Therefore the adversary knows both keys which are different and Alice and Bob never notice the messages have been modified.

11.6

Decryption: On input x and <c1,c2>. Compute $c1^x, compare\ if\ c1^x = c2$. If it is equal it means that $c2 = (g^y)^x = h^y = (g^x)^y$ so b=0 ; else b=1

Let's show this encryption scheme is CPA-secure by contradiction.

Assume it is not CPA-secure, it implies that there exists an adversary A such that for all negligeable functions : $\Pr\left[PubK^{eav}_{A,\Pi}(n) = 1\right] > \frac{1}{2} + negl(n)$

Let's construct an adversary A' for DDH. A' receives a DDH instance $(G, q, g, g^x, g^y, h')\ where\ h' = g^z\ if\ b = 1\ or\ h' = g^{xy}\ if\ b = 0$.

A' sends the challenge ciphertext <c1,c2>= <g^y, h'> with the public key pk = (G,q,g,h) $= (G, q, g, g^x)$ to A. Moreover the message space is {0,1} so A considers $m_0 = 0, m_1 = 1$. A' outputs whatever A outputs.

$\Pr[PubK_{A,\Pi}^{eav}(n) = 1] > \frac{1}{2} + negl(n)$

<=> $\Pr[A'(G, q, g, g^x, g^y, g^z) = 1] > \frac{1}{2} + negl(n)$

<=> $\Pr[A'(G, q, g, g^x, g^y, g^{xy}) = 1] > \frac{1}{2} + negl'(n)$

So $\Pr[A'(G, q, g, g^x, g^y, g^z) = 1] - \Pr[A'(G, q, g, g^x, g^y, g^{xy}) = 1] > \frac{1}{2} + negl(n) - \frac{1}{2} - negl'(n)$

<=> $\Pr[A'(G, q, g, g^x, g^y, g^z) = 1] - \Pr[A'(G, q, g, g^x, g^y, g^{xy}) = 1] > negl(n) - negl'(n)$

We know that $negl(n) - negl'(n)$ is also a negligeable function.

This contradicts the assumption that the decisional Diffie-Hellman problem is hard relative to G ($\Pr[A'(G, q, g, g^x, g^y, g^z) = 1] - \Pr[A'(G, q, g, g^x, g^y, g^{xy}) = 1] \leq negl(n)$ for every A')

Therefore the scheme is CPA-secure.

<u>11.7</u>

This scheme is not CPA-secure.

Notice that G is the quadratic residues of p so despite $h^r \in G$, $h^r + m$ *does not have to be in G*.

Let's construct an adversary A that chooses uniformly $m_0, m_1 \in \mathbb{Z}_q$. A gets <c1,c2>. Now A can know if $c2 - m_0$ is in G since it is easy to verify if a number is a quadratic residue. If b=0 then $c2 - m_0 = h^r \in G$ whereas if b=1 from our note earlier, $h^r + m_1 - m_0$ does not have to be in G. A verifies also if $c2 - m_1$ is in G. If only one of them is in G then A outputs this bit, otherwise if both are in G then A chooses randomly. By choosing randomly A succeeds with probability ½ but now each time $c2 - m_b$ is not a QR, which happens with half of the time ,A succeeds for sure. So A succeeds with probability $\frac{1}{2} * \frac{1}{2} + 1 * \frac{1}{2} = \frac{3}{4}$.

<u>11.15</u>

a) If N1,N2,N3 are not pairwise coprimes : there exists $i \neq j \in \{1,2,3\}$:
$\gcd(Ni, Nj)$ *is non trivial and* $\gcd(Ni, Nj)\,|Ni$. An adversary A can factor Ni=pq. A can easily recover $\phi(Ni) = (p-1)(q-1)$ and then compute $d = 3^{-1}[\phi(N)]$. A has recovered d so the adversary can recover $r = ci^d$ and compute H(r) so that $c4 \oplus H(r) = m$.

Else N1,N2,N3 are pairwise coprimes. An adversary A can use the Chinese Remainder Theorem to solve the system : $r^3 \equiv a1[N1], r^3 \equiv a2[N2], r^3 \equiv a3[N3]$ and find $r^3[N1N2N3]$.

We know $r \in \mathbb{Z}_{N1}^*$ so 0 < r < N1 <=> 0< $r^3 < N1^3 < N1N2N3$ since N1<N2<N3. This means that $r^3[N1N2N3]$ is equal to r^3 *in* $\mathbb{Z}$. Thus A finds $r = \sqrt[3]{r^3}$ and computes H(r) so that $c4 \oplus H(r) = m$.

b) To obtain a CPA-secure method we could choose uniformly 3 independent values $r1, r2, r3 \in \mathbb{Z}_{N1}^*$ and send <c1,c2,c3,c4,c5,c6> = <$r1^3[N1], r2^3[N2],\ r3^3[N3],\ H(r1)\oplus m,, H(r2)\oplus m,, H(r3)\oplus m$ > of length 3l+O(n).

c) We want a ciphertext of length $l + O(n)$. Let's modify our method from question b : choose a uniform $k \in \{0,1\}^n$ and choose an CPA-secure private key Encryption scheme. Send the ciphertext : < $c1, c2, c3, c4, c5, c6, c7$ > = <$r1^3[N1], r2^3[N2],\ r3^3[N3],\ H(r1)\oplus k,, H(r2)\oplus k,, H(r3)\oplus k,\ Enc_k(m)$ >

<u>12.1</u>

Consider $\Pi' = (\mathrm{Gen}', \mathrm{Sign}', \mathrm{Vrfy}')$ be a fixed length signature scheme. Let's construct $\Pi = (\mathrm{Gen}, \mathrm{Sign}, \mathrm{Vrfy})$ an arbitrary length scheme.

•Gen = Gen'

•Sign : On input $sk \in \{0,1\}^n$ *and* $m \in \{0,1\}^*$ *with* $|m| = l < 2^{\frac{n}{4}}$. Parse m into $m_1, \ldots, m_d$ blocks each of length $|mi| = \frac{n}{4}$ (pad the last block with 0's if needed). Uniformly choose $r \in \{0,1\}^{\frac{n}{4}}$. For each i=1,...,d compute $\sigma_i = Sign_{sk}'(r||l||i||mi)$ where i and l are encoded as strings of length n/4. Output $\sigma =< r, \sigma 1, \dots, \sigma d >$.

•Vrfy : On input $pk \in \{0,1\}^n$ and $m \in \{0,1\}^*$ and $\sigma =< r, \sigma 1, \dots, \sigma d >$

Parse m into d' blocks each of length $|mi| = \frac{n}{4}$. Output 1 if and only if d'=d and for each i=1,...,d , $Vrfy_{pk}'(r||l||i||mi, \sigma i)$ = 1.

12.5

a) $Vrfy(m, \sigma)$: On m run enc(.) then compare if $\sigma^e = (enc(m)^d)^e[N] \;\; =?\, enc(m)$. If it is output 1 else 0.

b) The no-message attack consists of choosing uniformly $\sigma \in \mathbb{Z}^*$ and computing $m = \sigma^e[N]$ so that (m, σ) is a valid forgery. However with this scheme it is impossible to compute m from σ. Indeed, $\sigma = enc(m)^d[N] <=> \;\sigma^e = enc(m)[N]$ but with an appropriate choice of enc() an adversary cannot decipher and thus cannot get a valid m.

c) ||N|| is publicly known so an adversary A can easily parse the suffix $0^{||N||/10}$. Moreover A also knows $l = ||m||$ so it is also easy to parse $0x00||m$. Therefore A can decipher and can thus do a no-message attack.

d) As in question c, an adversary A knows $l = ||m||$ so it is easy to decipher enc(m) by parsing 0||m||0||m, and then do a no-message attack.

Homemade question

A) $Vrfy_{sk}$'(m,t) : the receiver computes $t' = Sign_{sk}(m)$ (since Π is deterministic it will output the same if same input) and compares if $t' = t$. If it is equal then output 1 else 0.

B) If Π is unforgeable then $Pr[Sign - Forge_{A,\Pi}(n) = 1] \leq negl(n)$.

Mac_{sk}' = $Sign_{sk}$ of Π, so $Pr[Sign - Forge_{A,\Pi}(n) = 1] = Pr[Mac - Forge_{A,\Pi'}(n) = 1] \leq negl(n) = \varepsilon$

Hence it is impossible for an adversary to output a message with a valid tag with more than negligeable probability, so $\Pi'$is existentially unforgeable. Note that Mac' doesn't use pk so making it public doesn't change anything.

C) If pk is public, then upon receiving (m_0, t_0) and (m_1, t_1) ,where one of the tag comes from Mac' and the other is a random tag≠ Mac', an adversary can easily determine the validity of the messages using $Vrfy_{pk}$ from Π (since it's public). Therefore private key authentication does not imply private key encryption.