1.5

Using a chosen-plaintext attack, the adversary can obtain the encryption of text of its choice. For the shift cipher he can just encrypt any letter and observe the shift produced from the cipher of this letter. 1 plaintext gives him the unique shift used to encrypt any other message.
For the substitution cipher he can just encrypt the entire alphabet and observe the shift used for each letter. The alphabet of plaintext gives him the shifts used to encrypt any other message.
For the Vigenere cipher he can just encrypt the same letter t times (where t is the size of the key) and observe the shift produced at each index. The size of the key of plaintext gives him the shift of each index used to encrypt any other message. If the adversary doesn't know the size of the key then he can just encrypt the same letter a big number of times and look for a repetion of a pattern.

1.11

For any other language compute $S_\tau = \sum_{i=0}^{n} pi^2$ where n is the size of the alphabet of this language and pi is the reference frequency of the ith letter in this language. Then do the same steps as for english just using this S_τ and the reference frenquencies of the language .

Polish would be the hardest language to cryptanalyze as no letter has a significant higher frequence than the other (the maximum frequence is the letter "a" being only 8.9percent) .

Monogenere & ligenere

1)
Gen : Define the key to be k=$k_1 k_2 .. k_t$ where each k_i is a mono-alphabetic substitution, that is each k_i is a fixed one-to-one mapping on the alphabet.
Enc : Pass as inputs a key k and a plaintext $m = m_1 \dots m_l$ then apply for each m_j the corresponding k_i such that j=i+nt, $n \in$ N
Dec: Pass as inputs a key k and a ciphertext $c = c_1 \dots c_l$ then apply for each c_j the inverse of k_i where j=i+nt, $n \in$ N

2) The monogenere encryption doesn't preserve the probability distribution of characters in the plaintext since each character is no longer shifted but substitued so we can't study a pattern at every t based on frequencies to find k_i.

3) We can still use Kasiski's method to find repeating pairs/triples of common letters so that with a long message we can obtain a multiple of t. Indeed if the same letters appear at two different multiple of t they will be encoded the same (same k_i used and substitution is fixed).

4) $c = em + k\ [26]$
<=> $(c - k)e^{-1} = m\ [26]$

5) We can use Kasiski's method to find t. We know gcd(e,26)=1 so $e^{-1} \in \{1,3,5,7,9,11,15,19,21,23,25\}$. We can do an exhaustive search iterating through the possible values for e^{-1} while using the same method as for breaking Vigenere encryption for

k. We get k from probabilities and e from trying values until we find something that "makes sense".

Perfect and computational secrecy

Assume that a scheme (Gen_S,Enc_S,Dec_S) for s=1 or 2 or 3 is perfectly secret. Let's prove the (Gen_C,Enc_C,Dec_C) is also prefectly secret, that is:
$P[Enc_C(m)= <Enc_1(u), Enc_2(v), Enc_3(u\oplus v\oplus m)>] = P[Enc_C(m')= <Enc_1(u), Enc_2(v), Enc_3(u\oplus v\oplus m)>]$ from the definition of perfect secrecy.
We know that $Enc_1(u)$ is independent from m since u is random. Same for $Enc_2(v)$ as v is random. $Enc_3(u\oplus v\oplus m)$ is also independent from m as when doing an XOR if even one of the numbers is random and uniformly distributed over the whole range, and independent of the others, then the result will be random and uniformly distributed.
Futhermore we know that the selection of any two schemes is independent of m. $<Enc_1(u), Enc_2(v)>$, $<Enc_1(u), Enc_3(u\oplus v\oplus m)>$ and $<Enc_2(v), Enc_3(u\oplus v\oplus m)>$ are each independent of m because they involve a random parameter. Now for each pair if the last scheme missing to form the tupple of $Enc_C(m)$ is perfectly secret (that is s=3 for the first pair, s=2 for the second and s=1 for the last) then adding it to the tupple doesn't change the independence of m.
$P[Enc_C(m)= <Enc_1(u), Enc_2(v), Enc_3(u\oplus v\oplus m)>] = P[Enc_C(m')= <Enc_1(u), Enc_2(v), Enc_3(u\oplus v\oplus m)>]$
Since the whole tupple $<Enc_1(u), Enc_2(v), Enc_3(u\oplus v\oplus m)>$ is independent of m.

2.10

a) Consider m,m'∈ $\mathcal{M}$ such that |m| ≠ $|m'|$ and consider c = $k_{|m|} \oplus m$ for an arbitrary key we have $\Pr[Enc_k(m) = c] = \Pr[k_{|m|} \oplus m = c]$
and $\Pr[Enc_k(m') = c] = \Pr[k_{|m'|} \oplus m' = c]$
Now by construction, as c comes from XORing with m then |c|=|m|. Therefore the probability with m' has to equal 0 since c could not possibly be constructed from this xor.
Thus $\exists m, m' \epsilon M, \exists c \epsilon C, \forall k \epsilon K : \Pr[Enc_k(m) = c] \neq \Pr[Enc_k(m') = c]$
So by lemma 2.4 the scheme is not perfectly secret.

b) If |m|< l for any m∈ $\mathcal{M}$ then add as many zeros as necessary at the beginning of m until |m| = $l - 1$ and add a 1 before the real message m starts (0...01m) . Now proceed with the same scheme as explained in the question. This modified version is perfectly secret since $|k_{|m|}|$ is fixed (=l) and it now is the same sceme as the one time pad that we know is perfectly secret. Obvisouly adding zeros at the beginning did not modify m since when decrypting with the same key we will find back all these zeros and the first 1 encountered will be discarded, it only notifies when the real message m starts.

3.2

Consider an experiment where an adversary requests the encryption of $m0 \in \{0,1\}$ or $m1 \in \{0,1\}^{q(n)+2}$ where q(n) is the upperbound on the length of the ciphertext when Π encrypts a single bit. Thus the ciphertext produced by the encryption of m0 can be of size at most q(n), whereas the ciphertext produced by the encryption of m1 can be of size up to $q(n)^2 + 2q(n)$. By looking at the size of the ciphertext received the adversary can obtain more information on the message. The two upperbounds have a non negligeable difference therefore :

$$\exists a\ PPT\ adversary\ A, \forall negl(n): \Pr[PrivK^{eav}_{A,\Pi}(n)) = 1] > 1/2 + negl(n)$$

3.3

Consider Π = (Gen,Enc,Dec) a scheme satisfying Definition 3.8 for messages of equal length. Now let's construct a scheme Π' = (Gen', Enc', Dec') from Π that satisfies Definition 3.8 for message of unequal length.

Gen' is the same as Gen.

Enc' takes as input a key k and a message m. It creates a message m'=$0^{l(n)-|m|-1}1||m$ whith|m'|=$l(n)$. It then encrypts m' using Enc.

Dec' takes as input a ciphertext c. It decrypts c using Dec and then discards the first zeros and the first 1 it encouters and outputs the rest of the message.

Now lets' prove that Π' satisfies definition 3.8 for message of unequal length.

Assume there exists an adversary A' that can break Π' for messages m0' and m1'.

Let's show that this implies there exists another adversary A that can break Π for equal lengths messages (which would be a contradiction).

A takes the same messages m0' and m1' and concatenate them such that m0=$0^{l(n)-|m0'|-1}1||m0'$ and m1=$0^{l(n)-|m1'|-1}1||m1'$ (the same as Enc' does). m0 and m1 now have the same length so A ask to encrypt them. Since A' can guess b with probability higher than ½+negl(n) for Π', then A can also guess with this high probability for Π since we followed the same construction as Enc'.

This is a contradiction to our initial condition that Π satisfies Definition 3.8. Thus Π' must satisfy definition 3.8.