Question 1:

a) False; for any intrusion detection system, there must be a tradeoff curve between false positives and false negatives (as we saw in HW1). Clearly, there is a point where false positive rate is 0 (only report negative) and a point where false negative rate is 0 (only report positive) and everything in between, so it is not inherently easier or harder to get lower false positive or negative.
b) False; the beginning of a TLS session involves the client and the server exchanging a seed for a secret and generating a key for the session using the server's (and potentially the client's) private key. Once the secret for the session has been generated, it is used to encrypt all traffic exchanged between the server and the client. Because different clients have different secrets for their sessions, they will not receive the same set of bytes. In some cases, the server can even use a per session private key, as in the case of DHE-RSA and others, which would prevent comprise of recorded traffic even if the server's private key were discovered. Of course, any page with dynamic content or used the state of the session with the client would also result in a different set of bytes.
c) False; firewalls are not intrusion detections systems. They prevent intruders from ever getting to the point of communicating with their targets, so even if they use simple signatures they cannot be classified as a signature based IDS.
d) True; as mentioned for part c, firewalls are prevention not detection systems, so their logs are not relevant. Database logs, on the other hand, are too far down the application pipeline to be of any use (i.e. most useful metadata such as source ip, request type, etc… are stripped away by the time we get to the database).
e) True; DNSSEC digitally signs DNS data so that it cannot be forged or manipulated, thus preventing DNS cache poisoning from masquerading sources.

f) If Alice knows or suspects that Catie is looking at packets only one at a time, then Alice can split the payload that she intends to send on to Bob so as to have them each fail the regex that Catie is using to detect malicious packets. This can be done for instance by sending over just a few bytes at a time, preventing Catie from dropping the packets as individually they could be part of almost any payload.
g) Assuming no major advances are made in determining sources of Tor traffic, the source ip/port values in web logs would no longer be useful/relevant since they have no relationship to the real source. As such, attack detection would not be able to rely on grouping by source ips anymore, thus making it much more difficult.
h) By attempting to uncover true (non-spoofed) source IP address or detecting common attack pathways in the network, traceback can help detect and stop DDoS attacks. However, the different methods of implementing traceback have their limitations. For example, in link-testing methods, incoming links are tested to see which carry attack traffic. This could introduce a lot of network traffic overhead. Other methods, such as ingress filtering, require coordination between and cooperation from ISPs, which can be difficult.

i) The idea is to engineer mail that has lots of “off-color” terms together with words that occur with high frequency in the genuine communications between Fred and Mike. The two most obvious terms to use are Fred and Mike themselves, but additional examples of correspondence between the two can be mined for less common words which can make the DoS more effective. The best part about this scheme is that as soon as genuine mail from Mike begins to be rejected, all the rest of the terms in that document are further associated with spam.

j) Frog boiling attacks rely on thresholding to determine outliers in IDS and is greatly benefitted if the definition of “normal behavior” for the network is learned over time. In this case, an attacker can prevent detection by performing dangerous actions in such a way as to consistently operate below the threshold for detection. Over time, if the system learns about typical behavior by computing a streaming mean or variance, the definition of normal behavior for the IDS can even be engineered by the attacker, making the IDS worthless and “cooking” any hopes of detecting the intruder before they do real damage.