Thea Traw

1. Alice wants to send Bob a long message, and she doesn't want Eve to be able to read it. Assume for this scenario that AITM is impossible.

First, Alice and Bob use Diffie-Hellman to agree on a shared key K. Since AITM is not possible in this scenario, they can be confident that K is known solely by the two of them (due to the mathematical properties of D-H–that is, the discrete logarithm problem). Next, Alice will encrypt her message using symmetric encryption (as that is faster than asymmetric encryption and she has just established a shared key with Bob). Also, it's a long message, so it will have to be encrypted in chunks. So she sends C = AES(K, M) to Bob. Bob decrypts by performing AES_D(K, C). Given that Alice and Bob agreed on using a suitable block cipher mode (CBC), then Eve will be unable to decrypt and read the message sent (there will be no vulnerability to frequency analysis or attacks like that).

2. Alice wants to send Bob a long message. She doesn't want Mal to be able to modify the message without Bob detecting the change.

In this scenario, Alice wants to make sure that Mal cannot modify the message without Bob detecting the change, but she does not need to prevent anyone from reading it. So Alice does not need to encrypt her message M. Instead, Alice will send M || E(S_A, H(M)) to Bob. By signing the cryptographic hash of the message with her private key, Alice ensures that Mal will be unable to change the contents of M without Bob noticing.

If Alice just sent M || H(M), then Bob would be able to notice if the message got changed by non-nefarious means (bit flips in transit or the like), as then $H(M_{received}) != H(M_{sent})$. But Mal could send along any M' || H(M') and Bob would be none-the-wiser to the changes, and he would have no reason to believe that M' was not from Alice. But if Alice signs the hash, then anyone can read the hash transmitted (since P_A is public knowledge) but no one else could have signed it–and thus sent it–because only Alice has S_A. And so even if Mal wanted to change the message to M', there would be no way to sign H(M') in order to make Bob believe it was from Alice and thus unmodified. This is because Bob computes $H(M_{received})$ and checks if it equals $E(P_A, E(S_A, H(M)))$, and only if the message is unmodified will those two values be equal.

3. Alice wants to send Bob a long message (in this case, it's a signed contract between AliceCom and BobCom), she doesn't want Eve to be able to read it, and she wants Bob to *have confidence that it was Alice who sent the message. Assume for this scenario that AITM is impossible.*

First, Alice and Bob will agree on a shared key K using Diffie-Hellman. Since there is no threat of AITM, they can be confident that only the two of them know K (but Bob is not necessarily confident that it is, in fact, Alice who he is communicating with). So, Alice will need to sign her message (well, the hash of her message) with her private key and send that as well. So Alice will encrypt C = AES(K, M) and then send C || E(S_A, H(M)). And then Bob will read the message by computing $M_{received}$ = AES_D(K, C) and then ensure that $M_{received}$ is from Alice by checking that E(P_A, E(S_A, H(M))) equals H($M_{received}$).

Since only Alice has S_A, her signature on the hash of the message ensures Bob that she sent it. (Because being Alice is equivalent to having possession of S_A, in this scenario, as we assume that everyone has a correct copy of everyone else's public key and that everyone's private key is only known by them. Otherwise, this equivalency is not true, as either Mal could have S_A or put a fake P_A that is actually P_{Mal} and trick people that way (as then Mal would have S_{Mal} which would look like S_A). But in any case, here we trust that the holder of S_A is actually Alice, and so her signature with S_A means that she sent the message.) We also know that Eve cannot read this message, as only Alice and Bob know K and M is encrypted by AES and some block cipher method like CBC. Eve also cannot obtain M through H(M) because cryptographic hashing is a one-way function.

4. Consider scenario #3 above. Suppose Bob sues Alice for breach of contract and presents as evidence: the digitally signed contract (C || Sig) and Alice's public key P_A. Suppose Alice says in court "C is not the contract I sent to Bob". (This is known as repudiation in cryptographic vocabulary.) Alice will now need to explain to the court what she believes happened that enabled Bob to end up with an erroneous contract. List at least three things Alice could claim happened. For each of Alice's claims, state briefly how plausible you would find the claim if you were the judge. (Assume that you, the judge, studied cryptography in college.)

Claim 1: Alice believes that Mal found a hash collision C' such that H(C') = H(C). So, instead of receiving (C || Sig), Bob received (C' || Sig). Mal could do this because P_A is available to everybody, and so Mal could see that H(C) = E(P_A, Sig) = E(P_A, E(S_A, H(C))), and so Mal knew what hash value to match to find a suitable hash collision. So when Bob went to verify that the contract was, in fact, from Alice, he checked that E(P_A, Sig) = H(C), and it did. But Alice argues that Bob checked that E(P_A, Sig) = H(C'), which it did, and so he actually received C'.

The judge is not particularly impressed by this claim, as cryptographic hash functions are designed to make the chance of such collisions infinitesimal. But all the judge would need to do to check is to have Alice produce the actual C and H(C)...if they exist…and see if H(C) = H(C’). Case closed.

Claim 2: Alice believes that Mal tricked Bob into using a false P_A, which is under Mal’s control–that is, P_{Mal}. Mal then sends Bob (C || Sig), which is legitimate except that Bob’s check of the signature is actually H(C) = E(P_{Mal}, Sig) = E(P_{Mal}, E(S_{Mal}, H(C))). And Alice had no idea that any of this was going down!

The judge thinks that this would be a very silly defense for Alice to use if it were not true (as Bob is using P_A–or at least what he thinks is P_A and not P_{Mal}–as evidence), and so is willing to entertain the idea that Mal duped Bob. This should be quick to resolve: Alice provides her actual public key, and the court will see whether the P_A Bob gives as evidence matches the P_A’ that Alice submits. (This would potentially require further investigation, such as whether Alice has a valid/trusted certificate linking AliceCom to P_A’ or whether Alice had used P_A’ in other communications–in order to verify that she did not generate some random key pair as a fake alibi.)

Claim 3: Alice believes that the NSA has developed a way to crack RSA by means of quantum computers that can factor *fast*. And worse, Mal works for the NSA and used this highly-important, highly protected technology to acquire Alice’s secret key (by factoring n into p and q and then computing d). And then Mal sent Bob an erroneous contract (C || Sig) while pretending to be Alice and using S_A. And Alice did not know she was compromised until she was summoned to court.

The judge thinks that this is rather implausible (but does not actually know what the NSA knows). (The judge also thinks that this would have seismic implications on the cybersecurity world.) But if Alice’s secret key is compromised (through no malpractice of her own), there is not much that Alice can do about that now except for damage control and generating a new key pair. So the judge thinks this is likely a lie (…but what if it isn’t?).

Claim 4: Alice says that Mal stole her private key through skulduggery and then used it to send Bob a false (C || Sig). This skulduggery might have included breaking into Alice’s house and laptop and stealing her key; tricking her into revealing it under the guise of IT–Mal: “Oh, looks like there’s something wrong with your digital signature! Let me help! What is your secret key?”–and apparently Alice is very trusting; or Mal threatened Alice with bodily harm into giving up S_A. No matter what, Alice says that she was actually the injured party first. And Mal is responsible for grievances unto Bob and Alice both.

The judge thinks that this sort of case goes beyond cryptographic rulings. If Alice loses her secret key to Mal in a physical or verbal manner, then there's not much else that can be done. Alice should get a new key pair and perhaps pursue damages on her own.

5. For this scenario, suppose the assumption that everybody has everybody else's correct public keys is no longer true. Instead, suppose we now have a certificate authority CA, and that everybody has the correct P_CA (i.e. the certificate authority's key). Suppose further that Bob sent his public key P_B to CA, and that CA then delivered to Bob this certificate:

```
Cert_B = "bob.com" || P_B || Sig_CA
```

In terms of P_CA, S_CA, H, E, etc., of what would Sig_CA consist? That is, show the formula CA would use to compute Sig_CA.

Sig_{CA} is equal to $E(S_{CA}$, H("bob.com" || P_B)). Since only the CA has S_{CA}, it must be the CA that provided Sig_{CA}. Anyone else who has P_{CA} can then validate that "bob.com" and P_B are linked by checking that H("bob.com" || P_B) = E(P_{CA}, Sig) = E(P_{CA}, E(S_{CA}, H("bob.com" || P_B))).

6. Bob now has the certificate Cert_B from the previous question. During a communication, Bob sends Alice Cert_B. Is that enough for Alice to believe she's talking to Bob? (Hint: no.) What could Alice and Bob do to convince Alice that Bob has the S_B that goes with the P_B in Cert_B?

First, Alice will validate the certificate. (Because if it's a fake certificate, then H("bob.com" || P_B) != E(P_{CA}, Sig_{CA}), as Sig_{CA} = E(S_{CA}, H("bob.com" || P_B)) and only the CA has S_{CA} and could have generated Sig_{CA}.) If $Cert_B$ is valid, then Alice knows that "bob.com" and P_B are linked, as per the stamp (signature) of approval by the trusted CA. So, Alice needs to make sure that "Bob" really is Bob by ensuring that "Bob" has S_B. So Alice asks for Bob to sign H($Cert_B$) and send it to her. That is, Bob needs to send Alice E(S_B, H("bob.com" || P_B || Sig_{CA})). Then, when Alice decrypts the message by doing E(P_B, E(S_B, H("bob.com" || P_B || Sig_{CA}))) and checks that it equals H("bob.com" || P_B || Sig_{CA}), she can be sure that Bob has the S_B that goes with the P_B in $Cert_B$, because only the owner of S_B would be able to generate that signature.

7. Finally, list at least two ways this certificate-based trust system could be subverted, allowing Mal to convince Alice that Mal is Bob.

Subversion 1: Mal manages to trick the CA into signing (“bob.com” || P_{Mal}) because the CA did not perform sufficient checks to verify Bob’s identity. Then, since Mal has S_{Mal}, Alice will have no way to know that she is confirming that Mal has S_{Mal} instead of Bob having S_{Bob}. Alice will think she is communicating with Bob the whole time.

Subversion 2: Mal steals S_B from Bob (possibly physically, possibly obtaining through negligence on Bob’s part, possibly in some other under-handed way). Then, when Alice asks for Bob to verify that he has S_B, Mal will instead send the response signed with S_B and Alice will believe that Bob must have sent it, because he is supposed to be the only person with knowledge of S_B.

Subversion 3: Mal infiltrates/betrays the CA and signs certificates that shouldn’t be signed. That is, Mal has S_{CA} and can now sign whatever contracts that Mal wants to. So, Mal could make a contract (“bob.com” || P_{Mal} || Sig) and send it to Alice. Alice will verify that it is legitimate (which it is, because Mal signed the Sig with S_{CA}) and then ask for Bob (Mal) to sign $Cert_B$ with S_B (S_{Mal}). Mal will be happy to oblige, and Alice will be fooled into thinking that Mal is Bob.

Subversion 4: Mal disseminates a fake CA public key (and has the fake S_{CA}, of course) and tricks everyone into believing that Mal is a trustworthy CA. Mal can now generate any contracts that Mal would like. So Mal can make $Cert_B$ = (“bob.com” || P_{Mal} || Sig) and put it out into the world. And if Alice trusts Mal’s fake CA’s certificate, then Alice will validate it (which will work, because E($P_{CA\text{-}Mal}$, Sig) = E($P_{CA\text{-}Mal}$, E($S_{CA\text{-}Mal}$, H(“bob.com” || P_{Mal})))) and then ask to make sure that “Bob” is Bob having S_B. But Bob is Mal, and Mal has S_{Mal}. So Alice will believe that the signature must have come from Bob, who is the sole owner of S_B, but S_B was S_{Mal} all along, and Alice is tricked.