File: cryptography/scenarios.pdf
File owner: Lysander Miller
Collaborations: Discussed ideas with Kiri Salij.

Cryptographic Scenarios

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.**

   Steps:

   a) Alice and Bob use Diffie-Hellman to agree on a shared secret key K

   b) Alice sends AES(K, M) to Bob where M is the message Alice wants to send

   c) Bob decrypts Alice's message using AES and K. In other words, Bob uses AES_D(K, C) where C is Alice's encrypted message.

   Why it works:

   With Diffie-Hellman, Alice and Bob can agree on a shared key without Eve knowing what that key is. Then, using symmetric encryption, Alice can send an encrypted message to Bob where only Bob (the other holder of K) will be able to decrypt the message.

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.**

   Steps:

   a) Alice calculates C = E(S_A, H(M)) before sending Bob C concatenated with M (where M is the plaintext message Alice wants to send)
   b) Bob calculates H(M) and checks if H(M) is equal to E(P_A, C). If so, that means that Mal didn't modify the message. If H(M) is not equal to E(P_A, C), that means that Mal did modify the message.

   Why it works:

   Because Alice is the only person with S_A, she's the only person who can encrypt H(M) such that P_A can be used to decrypt it. In other words, because Mal doesn't have S_A, Mal cannot change M and change C such that H(M) == E(P_A, C). Thus, if H(M) == E(P_A, C), Bob can be confident that Mal didn't change anything. On the flip side, if if H(M) != E(P_A, C), Bob can be confident that Mal did change the message.

Note that we use H(M) instead of M because public key encryption is used in practice exclusively for short messages (e.g., to encrypt a hash function digest).

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.**

   Steps:

   a) Alice and Bob use Diffie-Hellman to agree on a shared secret key K
   b) Alice first calculates M' = AES(K, M). She then calculates X = E(S_A, H(M)). Then, she sends M' concatenated with X.
   c) Bob uses AES_D(K, M') to get M.
   d) Bob then checks if H(M) == E(P_A, X). If so, Bob can be confident that it was Alice who sent the message.

   Why it works:

   With Diffie-Hellman, Alice and Bob can agree on a shared key K without Eve knowing what that key is. Then, because Eve doesn't know what Alice and Bob's shared key K is, Eve won't be able to decode M' || X. Afterall, M' can only be decoded through AES_D(M', K). Additionally, while Eve can get H(M) from X (as Alice's public key is public), de-hashing something is not possible. Thus, Eve will not be able to decrypt Alice's message to Bob.

   Bob can also be confident that it's Alice who sent the message because only the holder of S_A would be able to calculate X = E(S_A, H(M)).

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.)

One thing Alice could claim happened is that someone, let's say Mal, was able to find S_A. Mal could have then used S_A to send Bob (C || Sig).

If I was the judge, I would find this relatively believable. Afterall, people are not always super careful with their passwords. Or, perhaps Alice saved S_A on one of those password management websites and the website got hacked. For example, LastPass – a common password manager – was hacked a year or so ago and it resulted in a massive data breach. No matter how Mal found Alice's S_A, though, if Mal had S_A, it would be incredibly easy to send Bob (C || Sig). Thus, I would find this relatively believable.

Another thing Alice could claim happened is that, along the way, the contract that Alice sent Bob was accidentally modified by something like a solar-flare.

If I was the judge, I would be thinking that, if the contract Alice sent Bob was accidentally modified enough that Alice would be able to say "C is not the contract I sent to Bob," it would be incredibly unlikely that we could perform AES_D(K, C), get an output we'll call G, and have H(G) == E(P_A, Sig). Thus, if H(G) == E(P_A, Sig), I would not believe Alice. If H(G) != E(P_A, Sig), I would be more likely to believe Alice.

Finally, Alice could claim that she sent Bob (C' || Sig) and then Bob took Sig from (C' || Sig) and concatenated it with a new contract C (which is encrypted using AES(C, K)). Thus, resulting in (C || Sig). Alice would then go on to claim that, while her signature is concatenated to this contract C, she didn't send Bob C, she sent him C'.

Assuming that C is AES(K, M) and Sig is E(S_A, H(M)) (like in my answer to problem three), if I was the judge, I could simply check if H(AES_D(K, C)) == E(P_A, Sig). If H(AES_D(K, C)) == E(P_A, Sig), then I wouldn't believe Alice. If H(AES_D(K, C)) != E(P_A, Sig), then I would check if H(AES_D(K, C')) == E(P_A, Sig). If so, I would believe Alice.

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.

    The formula CA would use to compute Sig_CA is Sig_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?**

    If Bob sends Alice Cert_B, that is not enough for Alice to believe that she's talking to Bob. After all, anyone could go to bob.com and download Cert_B. If Bob wants to convince Alice that he has the S_B that goes with the P_B in Cert_B, after sending Cert_B, he could send Alice E(S_B, H(Cert_B)).

    Let's refer to E(S_B, H(Cert_B)) as M. After Alice received both Cert_B and M, Alice would extract P_B from Cert_B before checking if H(Cert_B) was equal to E(P_B, M). If H(Cert_B) == E(P_B, M), Alice can be confident that Bob has the S_B to go with the P_B in Cert_B.

    This works because only someone with S_B would be able to generate E(S_B, H(Cert_B)). Thus, if Alice is talking to Bob and Bob sends E(S_B, H(Cert_B)), Alice knows that Bob has the S_B to go with the P_B in Cert_B.

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

    The first way the certificate-based trust system could be subverted is if Mal applies for a certificate from the certificate authority for bob.com with the public key P_M. If the certificate authority then decided to create and sign the following certificate (Cert_M = "bob.com" || P_M || Sig_CA), Alice would then send messages using P_M to Mal thinking she was sending them to Bob. This mistaken belief would be reinforced by the fact that Mal would have the S_M to go along with P_M.

Another way the certificate-based trust system could be subverted is if the certificate authority is not actually trustable and, for the right price, is willing to send a certificate (Cert_M = "bob.com" || P_M || Sig_CA) to Mal. Again, Alice would send messages using P_M to Mal thinking she was sending them to Bob and this mistaken belief would be reinforced by the fact that Mal would have the S_M to go along with P_M.