CS338 Assgnment 6

Kuo Wang

April 2022

1 Assumptions and Available Tools

I have access to these tools:

1. AES encryption, with:
   - $AES(K, M)$ uses key K to encrypt message M.
   - $AES\_D(K, C)$ uses key K to decrypt cypher C.

   Beware that:

   - Alice and Bob need to use the same key k.
   - Everyone use the same block cipher mode.
2. Diffie-Helllman key exchange procedure to share the key k.
3. SHA-256 the cryptographic hash function, with:
   - H(M) means to hash message M.

   The function can be used to:

   - Check file integrity
   - use as signature when digest is encrypted with private key
   - store password with password string modified with random string x followed by string "$".
4. Public key encryption algorithm E(K,M) and public and secret key pairs (P,S) for everyone involved.

2 Scenario 1

The need:
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 PITM is impossible.
Response:
Take these steps:

1. Alice and Bob use Diffie-Hellman to agree on a shared secret key K.
2. Alice encrypts her message M into cypher C with $AES(K, M)$ and sends it to Bob.
3. Alice sends C to Bob.
4. Bob decrypts Alice's cypher C with $AES\_D(K, C)$.

Why this works:

- We know that without PITM, Diffie-Hellman can certainly avoid anyone other than Alice and Boy getting the key. Therefore, only Alice and Bob has k.
- Consequently, the encrypted message can only be decrypted and read by Alice and Bob.

3 Scenario 2

The need:
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.
Response:
Take these steps:

1. Alice gets SHA256 hash digest, $H_M = H(M)$, of her message M.
2. Alice sends H_M to Bob.
3. Alice sends bob M.
4. Bob uses SHA256 to hash the message he received, M, into H_B. If $H_M == H_B$, then the message is intact. Else, the message is compromised by Mal.

Why this works:

- We know that the SHA256 hash of any data will be distinct even with a small change in the data. Therefore, we use it to check file integrity.
- Consequently, With the SHA256 shared, Bob can easily check the data integrity of what came from Alice. This fulfills the requirement.

4 Scenario 3

The need:
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 PITM is impossible.
Response:
Take these steps:

1. Alice and Bob use Diffie-Hellman to agree on a shared secret key K.
2. Alice gets SHA256 hash digest, $H_M = H(M)$, of her message M. Then, she creates a signature, $sig_A = E(S_A, H_M)$ with her secret key.
3. Alice encrypts her message M into cypher C with $AES(K, M)$.
4. Alice sends both C and S_A to Bob.
5. Bob will begin his work NOW. Bob deciphers C with $AES\_D(K, C)$ to get M'.
6. Then, Bob gets hash $sigReceived$ from $H(M')$, and tests if $sigReceived == E(P_A, sig_A)$. If yes, the message came intact from Alice. If not, the message is compromised.

Why this works:

- It is for the same reason explained in scenario 1 that the scheme with AES with Diffie-Hellman can avoid Eve reading the data. Therefore, that requirement is met.
- To check authenticity of the file, we use signature. We know that $E(P_A, sig_A) == sig_A == E(S_A, H_M)$ holds true only if the signature received is made with Alice's secret key, which only Alice has access to. Therefore, with this scheme, Bob can check authenticity.

5 Questions

1. **Question:**

   - Consider scenario 3 above. Suppose Bob sues Alice for breach of contract. Bob 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".
   - 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.

   Response:

   - Alice claims that Bob changed the contract after he received it. I will side with Alice's claim only when it is so verified: the digitally signed contract is signed by Alice with SHA256 hashing mechanism of the original contract. If Bob changed the contract, then the changed contract's SHA256 hash thus signature will be different. Alice's claim will be valid only if $Sig! = sig_A$ for Alice's signature Sig_A and signature presented by Bob Sig.
   - Alice claims that someone in the middle got the contract and passed it onto Bob. I will find this case very not likely. For a person in the middle to get the contract and edit it, he needs to get ahold of k, which is virtually impossible given that K is shared with Diffie-Hellman. Then, the scheme will work only if Bob fails to check the signature: had a person in the middle changed the message, Bob's $sigReceived == E(P_A, sig_A)$ check must fail since $sigReceived$ depends on SHA256 hash of the received message, which will change and alert Bob of the compromised message had Alice's message been changed by a PITM. In this case, I will find it likely that Alice is lying if she cannot prove that Bob did not do his due diligence in signature checking.
   - Alice claims that her shared secret key and private key have been leaked to a third party, Alice's claim could be valid, since a compromised key could defeat the entire scheme. However, Alice will still be held liable for not doing her part in keeping the transaction safe.

2. **Question:**

   - 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:

   $$Cert_B = "bob.com"||P_B||Sig_{CA} \tag{1}$$

   In terms of P_{CA}, S_{CA}, H, E, etc., of what would CA consist? That is, show the formula CA would use to compute Sig_{CA}.

   Response:

   - The equation is below:

   $$Sig_{CA} = E(S_{CA}, H("bob.com")) \tag{2}$$

3. **Question:**

   - 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 $Cert_B$?

   Response:

   - Alice ought to verify the certificate agency on the validity of the certificate, and check the message itself with P_B, E, and H if she can. If the verifications passe, then the message is from Bob.

4. **Question:**

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

   - The certification agency's private key S_{CA} could be leaked. This way, Mal can forge a signature using S_{CA} and pretend to be the certified Bob in sending Alice messages.
   - Mal can get of the private keys of Alice and Bob or use any ways to steal their identities. Then, Mal pretends to be Alice and Bob to cheat the CA or RA to get the valid certificates issued. To Alice, Mal has valid certificate from Bob; to Bob, Mal has valid certificate from Alice.