Advanced Web Security - Assignment 4A

December 2020

Questions

1. How would security in the OTR authentication be affected if the Diffie-Hellman values were encrypted using the recipient's public key (instead of being signed by the sender's private key)?
2. What problem in OTR is solved using the Socialist Millionaire Problem?
3. In Signal, assume you get access to a chain key $ck^{(x,0)}$ from Alice's device. What information are you able to get about Alice and Bob's conversation?
4. Is the Asymmetric Ratchet (AR) used in Signal deterministic? What are the security implication if AR is deterministic versus if it takes as input some random coins?
5. In Signal, in what way(s) can Alice check that she is messaging with her intended interlocutor?
6. How does the Signal protocol achieve perfect forward secrecy? How does this method differ from the way OTR works?
7. How long is the healing window in Signal? Why?
8. The main goal of authenticated encryption (AE) is to simultaneously provide message integrity and confidentiality. Assume you are given:

   - A semantically secure encryption scheme
   - An unforgeable MAC

   How can you construct an AE scheme using these two building blocks?

Answers

1. If the recipients public key was used to encrypt the Diffie Hellman values in the OTR authentication we will have a key exchange that is vulnerable to a man-in-the-middle attack, which in turn would break the authentication needed in OTR. This is why OTR implements a signature-based authenticated DH exchange, called SIGMA. SIGMA means SIGn-and-MAc, which decouples the authentication of the DH exponential from binding of key and identity. We do this because the point of the OTR is to not be able to identify who sent the messages, even if you were able to read them after getting one of the keys.

2. The problem that is solved in OTR using the Socialist Millionaire Problem is to make sure the parties communicating are sure they have the same secret.
   Alice and Bob wants to make sure they share the same secret x = y.

   $x = y = H(PK_A \mid PK_B \mid g^{x_1 y_1} \mid ''sharedsecret)''$

   The fingerprint is concatenated with a low entropy secret and the negotiated DH value, which is in turn hashed and used in the SMP protocol.

3. The info we are able to get about Alice's and Bob's conversation in Signal (assuming we get access to a chain key $ck^{(x,0)}$ from Alice's device) are info about the current message key as well as said above the chain key. This will in turn give them access to all of the input to the current asymmetric ratchet, which means the DH and Alice's secret ratchet and the root key of that level. Knowing the chain key at a certain stage in a level, also reveals all the subsequent chain keys and message keys in the same asymmetric ratchet level.

   This is because the asymmetric ratchet doesn't take any randomness, once we know one input we can get all subsequent inputs on the same level. On top of that, since we have the root key and Alice's ratchet key from that level (say level 1), it is also possible to reconstruct the DH value for the next level (say level 2), and therefor compute the output of the asymmetric ratchet for that level.

   As we can see, if an advisory has gained this much access, they are able to compute output on 2 layers of keys materials.

4. Asymmetric Ratchet is deterministic. The reason for this is because both Bob and Alice needs to be given the same output from the asymmetric ratchet KDF.
   $(mtpk_B^{rchsk_A^0} + mastersecret) > KDF = rootkey(0)$

   In the asymmetric ratchet, we feed the mid-term public key of Bob to the power of the Alice's secret ratchet key, and the concatenated master secret. This will in turn gives us the first root key. The "random" value is in actuality a DH value using one public and one secret key of Bob and Alice.

If we had used entirely random values, we would see that Bob and Alice would get different root key output, which would cause them to have different keys to encrypt with and in turn cause problems when decrypting.

5. There are a couple of ways that Alice can check she is communicating with the intended interlocutor. The simplest way is to pose a question you believe only you and the other participant will know the answer too. This is a way to make sure that the person you are talking to haven't been hacked or maybe someone has gotten hold of their phone or number.

   The signal application also has a way of scanning the QR code of people, as well as sharing identity keys. You can choose to verify users in the application, which can be done by scanning the QR code or sharing a long string of numbers which is a concatenation of their fingerprint. This can be done using another messaging app, phone call or being in the same location.

6. The Signal protocol achieves perfect forward secrecy by using the X3DH key agreement protocol. X3DH stands for "Extended Triple Diffie Hellman". The forward secrecy is obtained by using ephemeral keys. Keys that are used only once, then deleted to preserve the privacy.
   To be more exact, the forward secrecy is provided because Signal uses their KDF(key derivation function) to derive new keys for each asymmetric and symmetric communication. Which derives new keys for each message/communication based on DH values, a ratchet key and root key.

7. The healing window in Signal is 2 asymmetric ratchets. This is because if an attacker has gained access to one level, they will be able to compute the DH value for the next level, using the secret ratchet and root key from the level above. This gives then also access to the chain keys on the new level. But not the new ratchet key because it will contain some randomness.

8. We can construct an AE scheme using the following building block: - A semantically secure encryption scheme - An unforgeable MAC

   We first have to encrypt the plaintext. If we use a semantically secure encryption scheme, the plaintext could be mapped to multiple different ciphertexts, which is good as it will be harder to compare known plaintexts towards each-other. Then we use a hash function to create a hash of the ciphertext, which is used as our MAC, and then we send both the cipher text and the MAC at the same time. It's also important to remember to use different input keys for the encryption and the hashing function.

   A semantically secure encryption scheme that can be used, could be El-Gamal or perhaps Palliers encryption. As the task doesn't ask us how the encryption works or how the hasing function works, we will not go into details about this. An examples of a protocol that uses this scheme is IPSec and is called ”Encrypt-then-MAC” and is one of the most secure known Authenticated encryption schemes used today.