Alex Israels

Homework 3: Cryptographic Tools

1. Define various ciphers:
   1. Transposition cipher:
      1. This cipher encrypts plaintext by reorganizing the plaintext into a permutation of the plaintext according to some fixed algorithm.
   2. Substitution cipher:
      1. This cipher encrypts plaintext by substituting the plaintext for another according to some fixed offset to create a ciphertext.
   3. Product cipher:
      1. This cipher uses a combination of two or more transposition or substitution ciphers in order to make the encrypted ciphertext less susceptible to cryptanalysis.
2. Data Encryption Standard (DES):
   1. Electronic Code Book (ECB) mode: **TODO**
   2. Cipher Block Chaining (CBC) mode: **TODO**
   3. Attack against ECB:
      1. An attacker could rearrange the blocks of ciphertext so that the data wouldn’t come out in order on the other side.
3. Brute force DES keys:
   1. Keys per Second:
   2. Time for key space of :
4. Hash Function:
   1. Inputs:
   2. Outputs:
   3. Minimum *C*:
   4. Maximum *C*:
5. Lossless Compression:
   1. No, this cannot be used as a cryptographic hash function because it means that it is computationally feasible to find two inputs that correspond to the same output through the hash function.
6. Software dissemination:
   1. No, not entirely. It is possible that an attacker could find a way to change the source code to produce the same hash since both are nonencrypted. To maintain high confidence with this protocol the developers could also provide a digital signature on the source code to make sure that it has not been altered.
   2. For similar reasons as stated before, the user shouldn’t have high confidence that the code hasn’t been altered because it doesn’t have a digital signature from the developer. If only the binary were distributed it would be a lot harder but still not impossible, just improbable, that the code performs only the advertised function because it is a lot harder to alter a binary file to produce the same hash and still implement the functionality required to perform malicious actions.
7. Forward Search:
   1. Steps for Eve:
      1. Eve needs to know that the value of the key is between 1 and 10000.
      2. Then she needs to intercept the encrypted number on its way from Alice to Bob
      3. She can then encrypt all numbers from 1 to 10000 with Bob’s public key until she has found a match for the intercepted encrypted number.
   2. The number is still subject to a forward search attack. For example, if the number chosen was 9, a brute force attempt would be able to calculate it. However, to be fully exhaustive it would take Eve a very long time making the attack calculably infeasible. Therefore, it is still subject to a forward search attack, but one that is very unlikely to succeed.
   3. No, bob does not know that the key came from Alice because she is encrypting with his public key.
   4. Alice can first encrypt the number with Bob’s public key and then encrypt that with her private key. By encrypting with her private key Eve would be unable to do a forward search attack. Additionally, Bob can now confirm that the number came from Alice because he would need to use her public key and then his private key to decrypt the number.
8. One-time pad:
   1. It is a Vigenère cipher with a random key that is at least as long as the message it is encrypting.
   2. Hurdles for radio station:
      1. You only get a new number every 3 seconds so it would take a really long time to encrypt a sentence. Just 20 letters would take 1 minute before you could get a new random key.
      2. Radio can be received by anyone, so the key could be captured and tried on the message. An attacker would just have to shift the numbers over 1 at a time until they got a match. It would be hard but not impossible.
      3. How do you securely tell the intended receiver when to start and stop recording the numbers without someone intercepting that message and being able to decrypt your message?
9. RSA Algorithm:

Result = {1,162,181,64}