Name:

Computer Security: Homework 4 – Answers and Comments

Due September 28, 2009 on compass. Shortened late hand in period to October 2, 2009 to ensure adequate time to post answers before exam.

1. Alice and Bob use Diffie-Hellman to compute a shared secret. They select p=67 and g=13. Alice picks a k_{Alice} of 11 and Bob picks a k_{Bob} of 7.
   a. Show the computations for K_{Alice} and K_{bob}.

   $K_{alice} = g^{kalice} \bmod p = 13^{11} \bmod 67 = 38$

   $K_{bob} = g^{kbob} \bmod p = 13^{7} \bmod 67 = 2$

   b. Show how Alice and Bob use K_{Alice} and K_{Bob} to compute the shared secretK

   $kab = K_{bob}^{kalice} \bmod p = 2^{11} \bmod 67 = 38$

   $kab = K_{alice}^{kbob} \bmod p = 38^{7} \bmod 67 = 38$

   c. Which values of p, g, k_{Alice}, k_{Bob}, K_{Alice}, and K_{Bob} can be made public without affecting the security of the key exchange?

   K_{alice}, K_{bob}, p, and g can be made public.

   d. Explain how Eve could launch a replay attack against Alice and Bob.

   This is what I get for adding a section at the last minute. My attack was replaying Kalice and/or Kbob after figuring out kab. Of course this would only work if Alice or Bob was using the same private key value. This is very unlikely. Or if they were using the same private key, they are using the same kab, so there is no need to replay.

   A number of you described a man-in-the-middle attack. Not a traditional replay attack, but a valid attack.

   I gave full points in almost all attempts. A few of you tried to assert that the private keys were in danger, and that assumes the discrete logarithm had been solved.

   e. How could the the basic Diffie-Hellman exchange be augmented to protect against your replay attack?

   The messages that include the public keys could include some freshness information like a nonce or a timestamp.

   Some suggested using certificates for authentication, which would avoid a man-in-the-middle attack (assuming your certificate infrastructure was initialized correctly).

2. Work with Gnu Privacy Guard (GPG). You can access GPG from http://gnupg.org. I have used this on Linux and installed it via yum on my personal system. I am

Name:

running it on my Windows system via cygwin. It is already be installed on the University Linux systems. Type "man gpg" to check if it is installed. Once you get your GPG system operational perform the following tasks:

Most people who attempted this question got full points.

a. Create a key pair and submit your exported public key.

Full points assuming key file imports. Most

b. Sign the class public key posted at http://www.cs.illinois.edu/class/fa09/cs461/assignments/cs461-pub.asc with your key. Submit the exported signed key.

Imported the signed class key and used the –list-keys option to look at the signatures on the class key.

c. Select an ascii file. Encrypt it with the class key and sign it with your key. Submit the signed and encrypted file.

Used the -d option to undo the signature and the encryption. Depending on how the file was created, this took two -d steps.

3. Alice is posting her open source program. It is potentially going to be mirrored at other sites.

a. She uses DES-CBC to create a hash of her open source program files. What information would she need to make available for any person downloading the program to verify its integrity.

T*alking with one student, there was some confusion on how many bits of the last block to use. In the text, it simply says that n of the 64 bits in the last block could be used for a hash. I told him to use all 64 bits, and I answered this question in the newsgroup too.*

The verifier will need the 56 DES key used to create the hash, the hash value, and information about how the hash was created (if only a subset of the bits in the last block are used). The verifier may also need the IV. If the file is greater than two blocks, the IV isn't strictly necessary.

b. If Alice is worried about Eve launching a birthday attack, what order of number of hash tests would Eve have to perform to find a pair of messages that breaks the hash?

I was aiming for the birthday attack, so the answer should be O(2^64/2). Most answers were 2^32 or 2^33. Some people included a multiplier of 1.25 which I

Name:

believe is discussed in the handbook of applied cryptography. If they showed their work, this got credit too.

A few people were aiming for a specific exact hash match. Which would be more on the order of (2^64).

c. If Eve has access to a machine that can compute 10 million hashes per second, roughly how long would it take her to find a hash collision with better than 50% probability.

I was just aiming for them to get some feel for the real time hit. So it should be something like

*2^32/(10*10^6) = 439 seconds*

d. Alice changes to use MD5. How many operations would a birthday attack take?

MD5 is 128 bits, so the number of file variants should be O(2^(128/2))

e. Assuming Eve can still compute 10 million hashes per second, how long would it take for her to find a collision with 50% probability?

2^64/(10^7) = 58,494 years. A lot longer than the DES case.