

Information Assurance: Homework 5

Due October 13, 2006.

1. The strength of the RSA algorithm is based on the difficulty of factoring large prime numbers. Assume you are given the factorization for the modulus of a public key ($n = p \times q$). Show now this breaks the RSA key pair.
2. Suppose Alice and Bob have RSA public keys in a file on a server. They communicate regularly using authenticated, confidential messages. Eve wants to read the messages but is unable to crack the RSA private keys of Alice and Bob. However, she is able to break into the server and alter the file containing Alice's and Bob's public keys.
 - a. How should Eve alter that file so that she can read confidential messages sent between Alice and Bob, and forge messages from either?
 - b. How might Alice and/or Bob detect Eve's subversion of the public keys?
3. Thanks to the birthday paradox one can find collisions using the DES-MAC cryptographic hash function in 2^{32} messages. Alice wants to take advantage of that fact to make it swindle Bob. She has two contracts. One that Bob is willing to sign and another that Bob is not willing to sign. She needs to generate a version of each that has the same DES-MAC crypto hash. Suggest how she might do this. Hint: adding white space and combinations of characters with back spaces do not change the meaning of the contracts.
4. Work with Gnu Privacy Guard (GPG) or Pretty Good Privacy (PGP). They both implement the same protocols, but PGP uses proprietary encryption algorithms. You can access free trial versions of PGP from <http://pgp.com>. I have used the Windows version. You can access GPG from <http://gnupg.org>. I have used this on Linux and installed it via yum on my personal system. It may already be installed on the University Linux systems. Type "man gpg" to see. I will be evaluating your results on my Linux box using GPG. So if you use PGP, be sure to create a key using some combination of DSA and ElGamal (algorithms supported by GPG). Once you get your GPG/PGP system operational perform the following tasks:
 - a. Create a key pair.
 - b. Get your key signed by at least one other person. Submit an exported version of your signed public key.
 - c. Encrypt a file using the instructor's public key (at <http://www.cs.uiuc.edu/class/fa06/cs498sh/hw5/skh-pubkey.asc> with fingerprint 388E 7466 4DD3 390E 8F36 A535 474D 5DC9 4912 BF7E) and sign it with your key. Submit the signed and encrypted file.
5. In the Otway-Rees protocol, both a session id (n) and nonces ($rand_1$ and $rand_2$) are used. Are both really needed? Would the protocol be equally resilient if only the session id or the nonces were used? Explain why or why not.

