COMP 4109: Applied Cryptography
Assignment 3

February 14, 2005

For this assignment, you will need files available online at:

http://www.scs.carleton.ca/~soma/comp4109/assign3/

The answers to these questions are due March 2nd in class. Turn in paper copies either in class or at HP 5137. In addition, please email program code and outputs to Mohammad Mannan (mannan@ccsl.carleton.ca).

1. Describe an application in which a birthday attack would affect the security of a MDC. Describe another application in which birthday attacks are not significant. What is the key difference between these two scenarios?

2. PGP/GnuPG allow users to sign the keys of other individuals to establish trust relationships, forming the so-called "Web of Trust." By default, though, both programs create public keys that are "self-signed," in that the public key and associated metadata (name, email address, etc.) are signed using the matching private key. What is the purpose of such self-signatures?

3. Using PGP/GnuPG, if you encrypt a file for several recipients, each of whom has a different public key, you will find that the resulting file is only slightly larger than if you encrypted it for just one recipient. Why is this?

4. For these questions, use the test document and class public key available on the assignment webpage. Also, you will need to use pgpdump, either as a stand-alone binary or through a web interface. Note that you may want to refer to RFC 2440 for an explanation of pgpdump's output.

   (a) What is the DSA public key embedded in the COMP 4109 public key? Give all parameters in hexadecimal using the conventions of the textbook.

   (b) What precisely is the value of the DSA signature of the test document? Give all parameters in hexadecimal using the conventions in the textbook.

   (c) If a second signature was created of the same document using the same pubic/private keys, would you expect that signature to be identical to the one you already have? What if the class key was an RSA key?

   (d) Extract the DSA values from the test document signature and the class public key. Use these numbers to manually verify the signature using the DSA verification algorithm in the book. You will need to either create a program to

perform these operations (using a big integer library), or use a discrete math program that supports large integers (such as the UNIX `bc` command or Mathematica).

5. For these problems, refer to the description of RSA in Section 11.3 of the textbook. Assume that $R(m)$ (the redundancy function) is deterministic.

   (a) If you use the standard RSA signing algorithm to sign two identical documents, will the two signatures be equal?

   (b) Let $R(m)$ be the identity function: $m = R(m)$. Given two documents x and y with RSA signatures s_x and s_y (both created using the same private RSA key S), construct a third document with a valid S signature.

   Note that this is an "existential forgery" attack against RSA.

   (c) Assume that we are using a "secure" $R(m)$, such as the ones described in Sections 11.3.5 and 11.3.6. We have received a signature on an important document that has a multi-character error in formatting that nevertheless can be easily corrected.

      i. If the signature verifies correctly *before* correcting the one-bit error, do we accept the signature? Why?

      ii. If the signature verifies correctly *after* correcting the one-bit error, do we accept the signature? Why?