Coding and Cryptography Fall 2016
Mission #6
Due Monday, October 17th

Katherine E. Stange

October 3, 2016

Note: This mission is a two-week mission with individual components and shared components. It will count double. You can hand it in early or come to office hours on the day of handing in if you'd like feedback before the quiz on October 19th.

Individual work (phase one)

1. This mission is about the *continued fraction low exponent attack* on RSA. The attack works when someone has chosen a public/private key pair with a small decryption exponent d. Please review the workings of RSA, and in particular, the meaning of the usual symbols: $n = pq, de \equiv 1 \pmod{\phi(n)}$.
2. The first week, each team member will identify themselves as needing to work on coding (programming) or theory (proofs etc). You should choose the task which is more challenging to you. Your teammates or instructor can help you as you work through the task (consider using shared workspaces such as cloud.sagemath.org or overleaf or sharelatex).

If you are working on coding

1. You will individually code up the algorithms which perform the attack. There are two algorithms needed, and one will call the other, so you should divide up the work so someone is coding the *continued fraction algorithm* and someone else is coding the *attempted factorization phase*. The notation here matches your book on pages 104 and 171.
2. Here is a description of the continued fraction algorithm, which takes an input a single pair (e, n) of integers and outputs a sequence of pairs (p_n, q_n) of integers:
   (a) Take as input a pair of integers (e, n). Let $x_0 = e/n$.
   (b) Let $p_{-2} = 0$, $p_{-1} = 1$, $q_{-2} = 1$, $q_{-1} = 0$.
   (c) Compute the sequence of pairs (p_n, q_n) for $n \geq 0$ according to these recurrences:

$$\begin{aligned} a_n &= floor(x_n), \\ p_n &= a_n p_{n-1} + p_{n-2}, \\ q_n &= a_n q_{n-1} + q_{n-2}, \\ x_{n+1} &= (x_n - a_n)^{-1}. \end{aligned}$$

(d) If $x_n = a_n$, stop before computing x_{n+1}; the algorithm is done.

3. Here is a description of the attempted factorization phase, which takes as input a pair of integers (A, B) and a public key (n, e) and returns either False or a factorization of n:
   (a) Compute $C = (eB - 1)/A$.
   (b) If C is not an integer, return False. (Otherwise continue these instructions.)
   (c) Compute the roots of the quadratic polynomial
   $$X^2 - (n - C + 1)X + n$$
   using the quadratic formula.
   (d) If the roots are not integers, return False. (Otherwise continue these instructions.)
   (e) If the roots r_1 and r_2 are integers, then verify that $r_1 r_2 = n$ is a non-trivial factorisation of n. If so, return this factorization. Otherwise return False.

If you are working on theory

Your task is to read and understand Section 3.12 and Section 6.2.1 of the text. The first explains what continued fractions are. The second explains how the low exponent attack works using continued fractions. Ideally, one person can do one and the other person can do the other. For reading your section, do the following:

1. Take point-form notes while you read.
2. Reorganize the ideas into a brand new point form summary which probably doesn't match the order things were presented in the book, but matches your understanding. Emphasize the core idea, and don't include unecessary details.
   (a) For Section 3.12, cover: what is a continued fraction, what are the approximations it produces (i.e. defn of the p_n, q_n), what is the difference between continued fractions for rational and non-rational numbers, the main theorem (top of page 103), and the significance of this theorem as a statement about good approximations.
   (b) For Section 6.2.1, cover: the basic structure of the attack in outline, a precise quantitative statement of the fact that if d is small, k/d is a good approximation to e/n (don't give details of the proof, it's just a lot of inequalities, but state the relationship precisely), and explain, under this assumption, why the algorithm works.
3. Work out a small novel example by hand (calling on the computer to do any annoying computations if desired).
4. Now, with the book closed, but with your point form summary and example at hand, write up a LaTeX set of notes describing in full paragraphs the material, algorithm, and example. Write as if writing a text book for your peers (one that you would find easy to read).

Tasks as a group (phase two)

As a group, you will come together with expertise in coding and theory from your individual assignments.

1. First, the people who have worked on theory should give a presentation to their groupmates about what they learned. The other group members should take notes and ask lots of questions, as if this were a lecture. After the lecture, everyone should read the latex'd material and work on editing it together, keeping in mind their questions and confusions as a guide to making it better.
2. Second, the people who have worked on coding will explain what they have done and demonstrate the algorithms they produced. The rest of the group will examine them and test them, debugging if necessary.
3. Use the two algorithms to verify the example at the bottom of page 171. This will check your code is working or figure out which part is not working.
4. Now, as a group, put together the two algorithms to obtain the private key corresponding to public key $n = 160523347$, $e = 60728973$.
5. Hand in: LaTeX description of theory, the two algorithms, and a description of how you found the private key. Identify which group members had which roles (which part of theory or coding).
6. The final grade will be 1/2 group grade based on quality of the project as a whole and 1/2 personal grade based on the quality of your marked portion of the work in more detail.