Bleichenbacher

Description of Attack

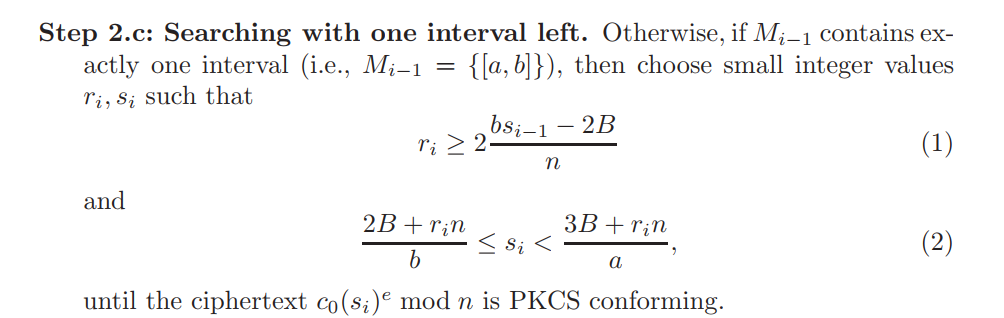
For each PKCS conforming s\_i, we update the set M of possible intervals where the message m is contained and narrowing the solutions.

We follow the steps specified in the paper exactly, first,

Step1: it is skipped since the cipher is already bounded.

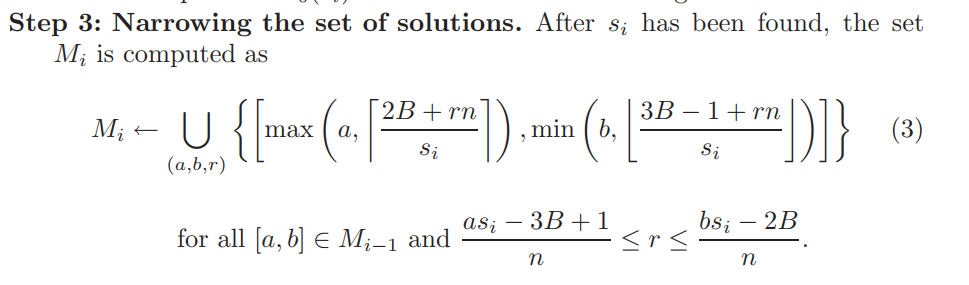
Step 2

1. If i=1: we find the first PKCS conforming s\_i starting from n/3B.
2. Else if i>1 and set M >1: we find s\_i > s\_i-1 that is PKCS conforming
3. Else if the set M only contains one interval, we find the s\_i according to these relationships.



Step 3

We narrow the set of solutions after we have found a s\_i from step 2, specifically, we will compute the set M as follows:



Step 4

If M contains only one interval of length 1, then we can compute the original plain text easily, if M=[a,a], then the message is a and we can recover the message being sent by extracting the data after the second x00 byte.

Otherwise, we update i = i +1, and go back to step 2 and repeat.

Reference: <http://archiv.infsec.ethz.ch/education/fs08/secsem/bleichenbacher98.pdf>

How to run Bleichenbacher

No special dependency needed

1. Run the server using the following command

python3 server.py -d [path to decryption\_key.txt] -n [path to modulus.txt]

1. Run bleichenbacher using the following command

python3 bleichenbacher.py -c [path to cipher.txt] -e [path to encryption\_key.txt] -n [path to modulus.txt]