Jennifer Shi (20662230)
J77shi j77shi@uwaterloo.ca CS458

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

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

Step 2.c: Searching with one interval left. Otherwise, if M_{i-1} contains exactly one interval (i.e., $M_{i-1} = \{[a, b]\}$), then choose small integer values r_i, s_i such that

$$r_i \geq 2\frac{bs_{i-1} - 2B}{n} \tag{1}$$

and

$$\frac{2B + r_i n}{b} \leq s_i < \frac{3B + r_i n}{a}, \tag{2}$$

until the ciphertext $c_0(s_i)^e \bmod n$ is PKCS conforming.

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 3: Narrowing the set of solutions. After s_i has been found, the set M_i is computed as

$$M_i \leftarrow \bigcup_{(a,b,r)} \left\{ \left[\max\left(a, \left\lceil \frac{2B + rn}{s_i} \right\rceil\right), \min\left(b, \left\lfloor \frac{3B - 1 + rn}{s_i} \right\rfloor\right) \right] \right\} \tag{3}$$

$$\text{for all } [a, b] \in M_{i-1} \text{ and } \frac{as_i - 3B + 1}{n} \leq r \leq \frac{bs_i - 2B}{n}.$$

Jennifer Shi (20662230)
J77shi j77shi@uwaterloo.ca CS458

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]

2. 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]