#### 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

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

and

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

until the ciphertext  $c_0(s_i)^e \mod n$  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

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

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

# 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

- 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]