this_is_not_lsb (category: Crypto　points: 162　solvers: 34)

Tired of difficult problems? Ok, I give you a simple LSB Padding Oracle problem. Ah, my magic has exploded... Sorry

`nc this-is-not-lsb.seccon.games 8080`

解法

RSA暗号は乗法準同型暗号であるので、未知の平文mに対応する暗号文cが与えられたとき、平文kmに対応する暗号文を求めることができる。公開鍵をN, eとして$c = \mathrm{encrypt}(m), c' = \mathrm{encrypt}(km)$とすれば、

$$c' = (km)^e \bmod N = k^e(m^e \bmod N) \bmod N = k^e c \bmod N$$

である。さて、`check_padding`函数が教えてくれるのは「引数cを復号した結果m_cをnと同じ桁数の2進数で表記したときに`0011111111`で始まるか」である。ではどのような平文に対応するcを入れれば`check_padding`函数がTrueを返してくれるのかを具体的に考えてみる。ただし乗法準同型性を使いたいので平文はkmの形のものだけ考える。

$$km = 0011111111\ldots \longrightarrow \frac{255}{256} \times 2^{s-2} \leqq km < 2^{s-2} \longrightarrow \frac{255/256 \times 2^{s-2}}{m} < k < \frac{2^{s-2}}{m}$$

であるので、例えば$k = \left\lfloor \dfrac{2^{s-2}}{m} \right\rfloor$とすればよい。実際のところ$m$は不明なので予測値を入れることになるが、それが真の値より大きいか小さいかによって振る舞いが異なる。即ちmの予測値をm_0としたとき、

$$\left\lfloor \frac{2^{s-2}}{m_0} \right\rfloor m \fallingdotseq \frac{m}{m_0} 2^{s-2} = \begin{cases} 0100000000\ldots & (m_0 < m) \\ 0011111111\ldots & (m < m_0) \end{cases}$$

mとm_0の比が1から大きくずれていた場合こうはならない(0100010010…や0011111001…のようになる)が、flagが`SECCON{`から始まることとバイト長が既知であるので、それに従った予測をする限り相対誤差の上限は約256^{-7}となる。つまり`check_padding`$\left(\mathrm{encrypt}\left(\left\lfloor \dfrac{2^{s-2}}{m_0} \right\rfloor m\right)\right)$は$m_0 < m$ならFalseに、$m < m_0$ならTrueになる。このことを使えば二分探索が可能となる。