pqpq (category: Crypto　points: 96　solvers: 112)

Warm your hands first. It's not obvious, right?

解法

$c_2 = (p-q)^e \bmod n$の$(p-q)^e$を二項定理で展開すると

$$(p-q)^e = {}_eC_0 p^e q^0 - {}_eC_1 p^{e-1} q^1 + \cdots - {}_eC_{e-1} p^1 q^{e-1} + {}_eC_e p^0 q^e = p^e - ep^{e-1}q + \cdots - epq^{e-1} + q^e$$

となる。ここで、末尾のq^eの符号が正になっているのは$e = 2 \cdot 65537$が偶数であるため。さて、このことから以下が分かる。

$$c_1 + c_2 \equiv 2p^e - ep^{e-1}q + \cdots - epq^{e-1} = p(2p^{e-1} - ep^{e-2}q + \cdots - eq^{e-1}) \pmod n$$
$$c_1 - c_2 \equiv ep^{e-1}q - \cdots + epq^{e-1} - 2q^e = q(ep^{e-1} - \cdots + epq^{e-2} - 2q^{e-1}) \pmod n$$

$n = pqr$であったから、$c_1 + c_2 \bmod n$がpの倍数であることは$c_1 + c_2$がpの倍数であることを、$c_1 - c_2 \bmod n$がqの倍数であることは$c_1 - c_2$がqの倍数であることを意味する。したがって$\gcd(c_1 + c_2, n) = p, \gcd(c_1 - c_2) = q$により$p, q$が求まり、$n = pqr$から$r$も求まる。素因数分解できたから後は通常通りの復号処理！とはいかない。eが偶数であり、$\varphi(n)$と互いに素でないからである。一旦$n = pqr, e = 65537, c = c_m$としたRSA暗号を解くと$m^2 \bmod n$が得られる。ここから$m^2 \bmod p, m^2 \bmod q, m^2 \bmod r$が求められるので、$m \bmod p, m \bmod q, m \bmod r$を求め、$m \bmod pqr$を求める。簡単に「求めて」などと書いたがその方法は少々複雑であるので1つずつ順番に説明する。

まずは$m^2 \bmod p, m^2 \bmod q, m^2 \bmod r$を求める方法。これは簡単で、$m^2 \mod n = X$とするとある整数$k$を用いて$m^2 = X - kpqr$と書けることから、$m^2 \bmod p = X - kpqr \bmod p = X \bmod p$である。$q, r$についても同様。

続いて$m \bmod p, m \bmod q, m \bmod r$を求める方法。mod 素数における平方根はTonelli-Shanksのアルゴリズムなどを使うことによって求まるのだが、その原理や証明は省く。参考文献1. にアルゴリズムが記載されている。なお、素数pと整数aに対して、$x^2 \equiv a \pmod p$の解の個数は(1以上p以下の範囲で)以下のようになる。

$a \equiv 0 \pmod p$の場合……1個 ($x \equiv 0$のみ)
aが平方非剰余の場合……0個 (定義より明らか)
その他の場合……2個 (解の和 $\equiv 0 \pmod p$)

証明は「$x^2 \equiv a \pmod p \Leftrightarrow x^2 - a$が$p$の倍数」を使えば易しい。本問の場合は最後のケースに該当するので、$\bmod p, \bmod q, \bmod r$で各2通り、総計8通りの「解の候補」が出ることになり、どれが本当の解であるかは文字列に変換してみるまで分からない。solver.pyの`sqrtp`函数では、平方根が2つ存在する場合そのうち一方を返すことにしている。

最後に$m \bmod pqr$を求める方法。よく「中国剰余定理によって求める」と言われるが、本来この言い方は正しくない。中国剰余定理は「整数$a_1, a_2, \ldots, a_k$および対ごとに素な整数$n_1, n_2, \ldots, n_k$に対して、$x \equiv a_1 \pmod{n_1}, x \equiv a_2 \pmod{n_2}, \ldots, x \equiv a_k \pmod{n_k}$なる整数$x$が(1以上$n_1 n_2 \cdots n_k$以下の範囲で)一意存在する」ことを述べたものであり、そのようなxを求める手法ではないからである(中国剰余定理の証明は解を明示するものが多いからそれを念頭に置いた表現かもしれない)。それはともかくとして、解を求めるこ

とを考える。後述の方法で$x \equiv a_1 \pmod{n_1}, x \equiv a_2 \pmod{n_2}$なる$x \equiv a_{12} \pmod{n_1 n_2}$を求め、$x \equiv a_{12} \pmod{n_1 n_2}, x \equiv a_3 \pmod{n_3}$なる$x = a_{123} \pmod{n_1 n_2 n_3}$を求め、…というように計算していくと、最終的に所望するxが求まる。つまり、k本$(2 \leqq k)$の連立合同式を解くことは2本の連立合同式を解くことに帰着できる。ここで、少々突飛に感じるかもしれないが、互いに素な整数n_1, n_2について$x \equiv a_1 \pmod{n_1}, x \equiv a_2 \pmod{n_2}$の解を$f(a_1, a_2)$とおいて函数$f$の性質を考えてみよう。まず、$X = f(a_1, a_2), X' = f(a_1', a_2')$のとき$f(a_1 + a_1', a_2 + a_2')$はどうなるだろうか。

$$\begin{cases} X + X' \equiv a_1 + a_1' \pmod{n_1} \\ X + X' \equiv a_2 + a_2' \pmod{n_2} \end{cases}$$

より$X + X' \equiv f(a_1 + a_1', a_2 + a_2') \pmod{n_1 n_2}$であることが分かる。また、$X = f(a_1, a_2)$のとき$f(ka_1, ka_2)$はどうなるだろうか。

$$\begin{cases} kX \equiv ka_1 \pmod{n_1} \\ kX \equiv ka_2 \pmod{n_2} \end{cases}$$

より$kX \equiv f(ka_1, ka_2) \pmod{n_1 n_2}$であることが分かる。まとめると、

$$f(a_1, a_2) + f(a_1', a_2') \equiv f(a_1 + a_1', a_2 + a_2') \pmod{n_1 n_2}$$
$$kf(a_1, a_2) \equiv f(ka_1, ka_2) \pmod{n_1 n_2}$$

端的に言えばfは線型であるということ。これら2つの関係式を使うと$f(a_1, a_2) = a_1 f(1, 0) + a_2 f(0, 1)$が分かるので、$f(a_1, a_2)$を求めることは$x_1 = f(1, 0), x_2 = f(0, 1)$を求めることに帰着できる。

$$\begin{cases} x_1 \equiv 1 \pmod{n_1} \\ x_1 \equiv 0 \pmod{n_2} \end{cases} \qquad \begin{cases} x_2 \equiv 0 \pmod{n_1} \\ x_2 \equiv 1 \pmod{n_2} \end{cases}$$

である。x_1はn_2の倍数であるので整数k_1を用いて$x_1 = k_1 n_2$と書け、したがって$k_1 n_2 \equiv 1 \pmod{n_1}$である。つまり、$k_1 = n_2^{-1} \bmod n_1$なので$x_1 = k_1 n_2 =$`pow(`$n_2$`,-1,`$n_1$`)*`$n_2$と求められる。同様にして$x_2$も求めることができる。

以上で$m \bmod pqr = m \bmod n = m$が求まったので文字列に変換してパディングを取り除けばflagを得ることができる。

参考文献

1. mod pでの平方根 - Zenn