\*prerequisite - Modular inverse, [Z\_n](https://en.wikipedia.org/wiki/Multiplicative_group_of_integers_modulo_n)

This level will focus on implementing an optimization for the original attack.

Again, [here](https://eprint.iacr.org/2012/417.pdf) is the article for those interested

2 (integral) changes:

1. Step 1 added - starting off with a smaller and more precise range for to be found (i.e. better )
2. Step 2 evolved - a more efficient search for .

Proposition-

Let be coprime integers (no common divisors) s.t.

Then if and mod are PCKS conf’ then .

So

Proof omitted. Appears in the article. We call such a pair, a good pair.

Thus by finding such pair we infer i.e. we find more constraints.

Can be shown that pairs are better when is smaller. Moreover, bigger will make a greater impact and enhance the narrowing .

We’ll implement the following idea:

Find good pairs (“trimmers”)

Calculate (as if )

Find are good.

We’ll binary search these .

Then set

We’ll find the “trimmers” by:

Iterating over and check if such that (u,t) is good. If yes is good and move on.

“Skipping holes” -

Can be shown that values in can’t generate a PCKS conf’ message.

Intuitively, we need to move from [2B,3B-1] to [n+2B,n+3B-1], and from [n+2B,n+3B-1] to [2n+2B,2n+3B-1] and so on…, and they fall outside of this range.

Pseudocode:







