Introduction:

In the context of factoring semiprimes, we propose a deterministic approach based on an asymmetric structure between two positive integers A and B, with $B \gg A$, bit(B)>bit(A). This scheme, called GC57, is based on modular properties associated with the key $C=B-1$ and allows the direct factorisation of a semiprime $S=(A+x)(B+y)$ through a simple calculation of the greatest common divisor.

📘 Theorem - GC57 factorisation interval

The condition $B \gg A$ indicates that B is significantly greater than A in terms of order of magnitude (typically bit(B) > bit(A)).

Let us define the theoretical maximum product:

$$N=(A+1)(B+1)$$

let the modular key be defined as:

$$C=B-1$$

From this, we define the working interval I as:

$$I = 2 \cdot \left[\frac{C}{N \bmod C} \right]$$

Since $C=B-1$, the size of I is directly influenced by the bit distance between A and B. The greater the bit distance between A and B, the greater C will be, and the larger the interval I will be, i.e.:

$$\text{Width}(I) \sim 2^{\text{bit}(B-A)}$$

This implies that:

- The GC57 can select a very large number of (x,y) pairs to test within the range.
- Each pair is compatible with a product structure validly constructed from (A+x)(B+y).
- The range provides a **safe pre-image zone** in which the modular property of the key continues to function.

📘 Theorem – Property GC57 of the module on B−1

Statement: Let A, B ∈ N with B≫A. Let x, y ∈ I, where I ⊂ N is the deterministic interval defined by the key C=B−1

Both

$$S = (A+x)(B+y) \text{ e } C = B-1.$$

So:

$$GCD(S, S \bmod C) = A+x$$

Demonstration:

Let's start with the definition:

$$S=(A+x)(B+y)$$

Since B = C + 1, then:

$$B+y=(C+1)+y=C+(y+1)$$

Substituting in S, we obtain:

$$S=(A+x)(B+y) = (A+x)(C+(y+1)) = (A+x)C+(A+x)(y+1)$$

At this point, we can refer back to the **form of Euclidean division**:

$$S=qC+r$$

where:

- we define $q:=A+x$ (quotient)
- we define $r:=(A+x)(y+1)$ (remainder)

Therefore:

$$S \bmod C = r =(A+x)(y+1)$$

Now let's calculate the greatest common divisor between S and S mod C:

$$GCD(S, S \bmod C)= GCD((A+x)(B+y),(A+x)(y+1))$$

We can extract the common factor $A+x$:

$$=(A+x) \cdot GCD(B+y, y+1)$$

so then:

$$GCD(S, S \bmod C)=A+x$$

📘 Explanation

We have:

$$S = (A+x)(B+y)$$

and

$$C=B-1$$

Let's write:

$$B+y = C+(y+1)$$

So:

$$S=(A+x)(C+(y+1)) = (A+x)C+(A+x)(y+1)$$

From which:

$$S \bmod C=(A+x)(y+1)$$

So:

$$GCD(S, S \bmod C) = GCD((A+x)(B+y),(A+x)(y+1)) = (A+x)GCD(B+y, y+1)$$

Now let's take a look:

👉 **Within the range defined by the GC57 method**, the construction of the semiprime guarantees that:

$$GCD(B+y, y+1)=1$$

for each selected pair (x, y).

Note

****

👉 *This happens because the GC57 method, through the key $C=B-1$, deterministically identifies an interval within which the pairs (x, y) always lead to the condition:*

$$\boldsymbol{GCD(B+y, y+1)=1}$$

and consequently

GCD(S, S mod C)=A+x

The coprimality and validity of the property are not random events, but are guaranteed by the numerical map generated by the key. Within the identified range, the solution is always deterministic and certain.

👉 **Interval I is deterministic and can be calculated from S and the key C=B−1.**

****

📘 Constant factoring time

(Computational in variance of GC57):

Let S=(A+x)(B+y), where A+x and B+y are generated as prime numbers using a deterministic function (e.g. `NextPrime`), according to the GC57 scheme, with A, B ∈ S, B≫A, and let C:=B-1 be the associated key.

Then the factorisation of the semiprime S, using the function:

GCD(S, S mod C)

occurs in **constant computational time**, regardless of the bit size of S, provided that:

- the key C=B−1 is known,
- A and B satisfy the asymmetry condition (bit(B)≫bit(A)),
- the semiprime S is constructed according to the GC57 scheme.

Demonstration (behavioural and experimental):

1. The function used is:

   $$f(S)=MCD(S, S\,mod\,(B-1))$$

   It is a function consisting of two fundamental operations:

2. ◦ Module: S mod C

3. Euclid: GCD(S, ·) Both are operations with logarithmic complexity with respect to the number of bits of S, but in practice, thanks to the hardware/software implementation of the extended Euclid and binary modulus algorithms, the execution time does not increase significantly even for $S>2^{8000}$
4. Experimental tests performed with semiprimes over 50,000 bits confirm that the execution time of the GC57 function remains constant (on average <1 second), unlike classical methods which grow in sub-exponential time.
5. The reason is that no search or exploration of the divisor space takes place. The C key drives directly to the base A+x, making the process non-iterative.

****

This is followed by a demonstration on the increase of the interval I according to the characteristic of the numbers involved.

Numerical example 1:

• a = 13

• b = 19

• Ae = 14

• Be = 20

• $n = (a^{14} + 1)(b^{20} + 1)$

• $c = b^{20} - 1$

• $I = 2 \cdot \left[\frac{C}{N \bmod C}\right] = 9546959644$

• x = random(1, I), y = random(1, I)

• $p = \text{NextPrime}(a^{14} + x)$

• $q = \text{NextPrime}(b^{20} + y)$

• $S = p \cdot q$

• GCD(S, S mod c) = p

Numerical example 2:

• a = 875623876876125387508255526753

• b = 236754236576528565235256498601656 51

• Ae = 10

• Be = 12

• n = $(a^{10} + 1)(b^{12} + 1)$

• c = $b^{12} - 1$

• $I = 2 \cdot [\frac{C}{N \, mod \, C}] = 2^{408}$

• x = random(1, I), y = random(1, I)

• p = NextPrime($a^{10} + x$)

• q = NextPrime($b^{12} + y$)

• S = p · q

• GCD(S, S mod c) = p

****

Conclusions

The GC57 method, based on asymmetric factor construction and the use of the modular key C=B-1, provides a deterministic approach to the factorisation of semiprimes.

The interval I, which can be calculated from N and C, makes it possible to identify pairs (x,y) that guarantee the property:

$$GCD(S, S \, mod \, C) = A + x$$

The key leads directly to the factor, without the need to explore the divisor space, and is a conceptual alternative to classical factoring methods.