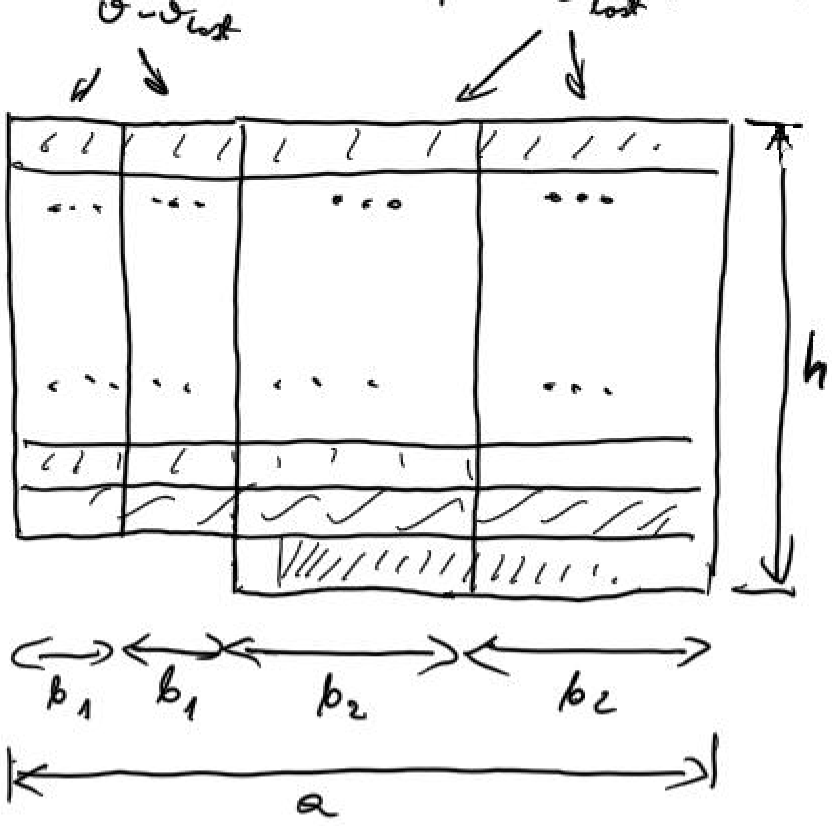
In the result we have two widths of (sub)block $e_{i,j}$:



Lee & Lim noted that better performance is achieved if $b_2 \geq b_1$

Then we need storage of size

$$(2^h - 1) v_{last} + (2^{h-1} - 1)(v - v_{last})$$

and the cost of exponentiations on-line (i.e. when we learn the exponent e) equals:

- $b_2 - 1$ squarings (we assume $b_2 \geq b_1$)
- $b_1 \cdot (v - v_{last}) + b_2 \cdot v_{last} - 1$ multiplications in the worst case

and:

$$\frac{2^{h-1}-1}{2^{h-1}} b_1 (v - v_{last}) + \frac{2^h - 1}{2^h} b_2 \cdot v_{last} - 1$$

multiplications on average.

In the paper "Improving and extending the Lim/Lee Exponentiation Algorithm" the authors note that the second version of the Lim/Lee method is never worse than the first one.

In the above paper we find a new version of the Lim/Lee algorithm:

- first, the number of input parameters (h, v, v_{last}) is reduced to two (a, b) and the dependency between parameters is also changed;
- secondly: optimization of pre-computation process is also considered.

"The dependency changed" means that for input a, b the number h of blocks e_i is computed as:

$$h = \left\lceil \frac{l}{a} \right\rceil$$

l — length of exponent e

and the number v of subblocks $e_{i,j}$ is obtained from the formula:

$$v = \left\lceil \frac{a}{b} \right\rceil$$

Then the number of bits in the last row is

$$a_{last} = l - a(h-1)$$

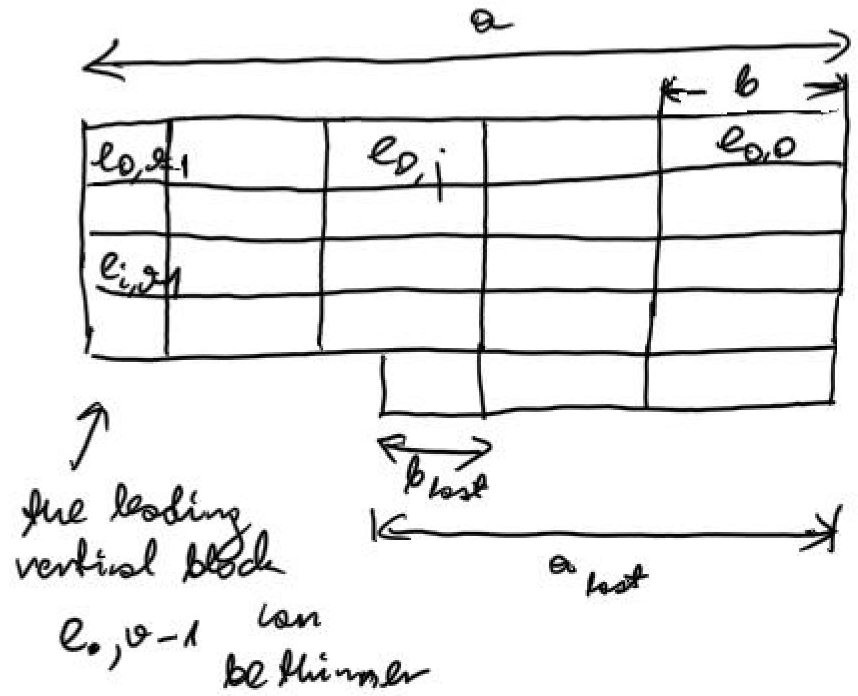
and they are divided into

$$v_{last} = \left\lceil \frac{a_{last}}{b} \right\rceil$$

(sub)block $e_{i,j}$. The number of bits in the last block is

$$b_{last} = a_{last} - b\,(v_{last} - 1)$$

The resulting picture seems to be almost the same as the first version of the Lee/Lim algorithm, but the parameters are calculated in a different way, and the last version according to the experiments done by its authors is never worse than the two original versions of the algorithm:



Moreover, having only two input parameters (a, b) it is easier to establish (by e.g. the brute-force search) the optimal ones (a_{opt}, b_{opt}) for a given bitlength l and storage limitations (expressed e.g. as the maximum number of group elements to be pre-computed and stored in memory). The cost of online exponentiation is

- $b-1$ squarings
- $\frac{2^{h-1}-1}{2^{h-1}}(a-a_{last}) + \frac{2^h-1}{2^h}a_{last} - 1$

multiplications on average with storage cost:

$$(2^h-1)v_{last} + (2^{h-1}-1)(v-v_{last})$$

precomputed elements.