Let $G = \mathbb{Z}_M$ be the group of units modulo M where $M = 25195908\ldots$ is the RSA-2048 challenge number (assumed to be of unknown factorization). Let G be the cyclic group of unknown order generated by a generator $g \in G$.

Consider a toy problem: let $N = p_1 p_2 \ldots p_k$ be a product of a large number of primes. We wish to compute the quantities g^{N/p_i} for each i.

The relation of this toy problem to plasma cash using RSA accumulators is as follows: first, allow exponents as in $N = \Pi p_i^{e_i}$, including zero exponents. Then g and g^N are accumulated hash values ("accumulator values") immediately surrounding (say) 100 blocks. Inclusion and exclusion proofs can be made showing that given g and g^N as "pbulic inputs", p_i factors into Ne_i times; this is much smaller in size than providing 100 merkle inclusion/exclusion proofs. One example of an inclusion proof is to provide g to the cofactor $N/p_{e_k}^i$, i.e. to provide $w = g^{N/p_{e_k}^i}$ and have the verifier check that $w^{p_i^{e_k}} = g^N$. For k inclusion proofs, the naive solution involves k modular exponentiations to a large number (the cofactor is almost the same size as N itself); but it is clear that these k modular exponentiations all share a large amount of substructure which we can exploit (indeed, the toy problem turns out to be solvable with $\log k$ modular exponentiations with exponents of size similar to N). This is only a toy problem because it doesn't generalize to weseolowski's proof of knowledge of exponent scheme or to exclusion proofs; the "target" that one proves knowledges of exponent of is not the same.

Solution to the toy problem: we do some precomputations. Set

$$B_0 = g^{p_{k/2+1}\cdots p_k}$$
$$B_1 = g^{p_1\cdots p_{k/2}}$$

we treat B_0 as "g raised to the cofactor of the leftmost half of the list of primes" and B_1 as "g raised to the cofactor of the rightmost half of the lits of primes". In the next round we compute four B-values $B_{00}, B_{01}, B_{10}, B_{11}$, each of which is g raised to the cofactor of a quarter of the list of primes. For e.g., B_{01} is g raised to the cofactor of $p_{k/4}\ldots p_{k/2}$, i.e., to $p_1\ldots p_{k/4}p_{k/2}\ldots p_k$, which can be calculated as $B_0^{p_{k/4}\cdots p_{k/2}}$. Each level of computation has the same total cost (since modular exponentiation is linear in the exponent size, i.e. linear in the log of the exponent). After $\log k$ such computations, we are done.