APPENDIX III

My BIC Arrays Intersection Solution

*** ACTUAL ***

Any one of the pdf's in the BIC folder contains the mathematicians' intersection solution. Once the math and their method are understood by any of you, apply the appropriate structure by altering my xyz structures so that their math solves the intersection on this algorithm.

The structures I have is, this is a count of 1's on a key/value pair, as anything on a computer, even RAM and Disk, is a key/value pair, even pointers. Maintaining on insert a unique key, when the intersection is applied to the key field, the numbers are reduced to only 1 or 0 at each position, the result being the original key/value pair.

So, for example, if I have the key value pair 010 – 111, I change this to an array of uint64, and for x, at position 1, I insert into the 0 bucket this array, at x position 2 I insert into the 1 bucket this array, and at x position 3, I insert this array into that 0 bucket. Y would be inserted into bucket 1 (01), bucket 2 (10), and for z, I would insert this array into bucket 010 at z position 1.

The insert is then simply a ++ to all the numbers at these positions.

So if I had 010 – 111, 011 – 010 k/v pairs; it would look like:
x: pos1: 0: 0 2 1 1 2 1, 1: 0 0 0 0 0 0; pos2: 0: 0 0 0 0 0 0, 1: 0 2 1 1 2 1;
pos3: 0: 0 1 0 1 1 1, 1: 0 1 1 0 1 0.
And so forth.

Authors may cite k intersecting subsets. This would be stored as k(x1,x2), k(x1,x3), k(x2,x3); which is easily done on insert.

The result is, in its most basic form, using uint64_t as your number sizes at each binary position, we can store exabytes within a matter of a few megabytes. The decompression speed is sufficiently fast, but is the slow operation. This then also creates a naturally ordered array of key/value pairs, making it the best possible data structure and algorithm, making trees, hashing, etc. obsolete. The algorithm easily forms a cube making in memory databases easily possible – where everything is indexed, even relations, so long as you follow what I said on that topic. The draw-back to this algorithm, and the reason I never completed it, is that once it is out there in the community, I was watching quantum computers break RSA by bulk bitwise between gates. This easily does the same on classical computers, and worse, bulk math between gates is possible with this algorithm. For bulk math, the challenge becomes knowing how to delete bitwise math operations once performed, to maintain data integrity in the correct binary buckets. If that is done, then yes, this creates then rainbow tables on all possible PKI-based cryptosystems, and also a rainbow table on the plain-text/cypher-text problem of symmetric key. I was absolutely right about this, that the only remaining crypto is AES used as One-Time-Pads.

For this latter reason, I never completed this algorithm.

It is my understanding that I was wrong, and the NSA doesn't actually have this, based on complaints by the FBI that they cannot break the cryptography that say, white militias are using.

But yes, Paul Allen did make up a peronie's disease joke article on me, under my full name, online, and yes, mark zuckerberg then started putting tape over his cameras like I started doing after I saw that article.

I was further going to use this algorithm to completely revise the computer, where all computations are done in bulk system-wide, the results decompressed by a massively parallel secondary graphics card – the result being all possible programs are on your disk instead of an app store, and load into the compressed OS as compressed software (think a floppy disk is software inserted into the computer which is the OS). This would then eliminate the need for RAM, the CPU falls back to being a brain for system calls, and the primary graphics card is for your graphics, unless you have the CPU do the graphics for you. The requirement was a secondary graphics card for the vastly increased on-chip L1/L2 cache, and the fact of having thousands of compute units instead of the 3 compute units of the CPU.

This algorithm is the best-possible algorithm for computing.

I told the NSA, Microsoft, and the world about this, while pointing out it requires the combinatorics intersection solution.

And they still do not have it...