Sep 25, 2016 at 3:08pm

[Quote](https://cicada3301.boards.net/post/55/quote/34)[like](https://cicada3301.boards.net/thread/34/solving-hard-little-information-failures)

Post Options

**Post by mortlach on Sep 25, 2016 at 3:08pm**

Introduction   
  
Apologies for the long post (it could have been much longer!) An aim is to make things understandable by all, with something for everyone interested, often this means omitting things, or explaining more simple concepts.  
  
  
TL;DR ? The puzzle is difficult, so difficult even knowing you’re wrong is difficult :) . High transactional costs prohibit collaboration and so individuals must *Believe nothing [from this book] except what you know to be true. Test the Knowledge. Discover your truth….* This is all probably deliberate by the puzzle makers.  
  
  
It is often asked about this puzzle: Why isn’t there a list of everything that has been tried? The attitude is often exasperation and frustration: How can so many supposedly smart people have spent so much time trying so many things and not reached (m)any firm conclusions? These are excellent questions and understandable sentiments. We will try and present some reasons why even though it is simple to ask such questions it is not so simple to adequately reply. Particularly, we will try and give an overview of why it is hard and why there is not so much information on failed decrypting attempts. The first reason is: The puzzle is difficult, in many ways. We don’t actually know what we’re supposed to do except in vague terms, we have 3 possible starting points:  
  
    A We have to find the deep web page  
  
    B We have to decrypt the runes  
  
    C We have missed something  
  
  
    [cicada3301.boards.net/thread/32/3-ways-progress-liber-primus](http://cicada3301.boards.net/thread/32/3-ways-progress-liber-primus)  
  
  
Just from that simple starting point the number of possible things to try grows fast. This post will focus on B – but think about C: if it is true then there are practically an unlimited number of things that are in the set of missed things.  
  
  
Decrypting the Runes  
  
  
There are many possible methods to encrypt a message and hence decrypt a message. Therefore, it would seem reasonable that we could just list possible methods then starting with Method 1, try all possibilities, if we don’t find ‘The Answer’, cross Method 1 of our list and then try all possibilities for Method 2, then if we still fail cross Method 2 off our list and try Method 3…etc. Although this is reasonable what we actually find it it’s very difficult to “try all possibilities” which then makes it even more difficult to “cross Method X off.” By now, you may be shouting “obviously trying all possibilities is such a huge space you won’t manage it, but surely you can make reasonable cuts to that space and try those, smaller sets of possibilities.” Well, sure, but what we hope to explain is why that is still difficult. Throughout the following the assumptions that have been made that can be reasonably questioned are highlighted, to help show how many assumptions we generally make (the first one being ***we can decrypt the runes...***)    
  
  
General Overview of Encrypting / Decrypting  
  
  
First let’s try and explain encryption in general terms. We have a message-text, MT, which can be written in a list as:  
  
  
    MT = F, O, R, A, L, L, I, S, S, A, C, R, E, D  
  
  
This message can be represented by numbers, for example the position, or value of the letters in the Gematria. There are also many other common ways to encode letters as numbers, generally we assume the Gematria is our starting to point, ***but there is no reason this must be the case.***  
Encoding a Message as Numbers from the Gematria  
  
  
**The ordering principle:*w****e will assume from the title of the Gematria that the order of the runes is important and should be preserved.* From previous pages we know that the runes have been ordered in ascending (Forward) or descending (Reverse) rune-prime-value. Applying the ordering principle reduces the sets of possible numbers for the runes to 59 distinct cases (a prime and also the same number of characters in the non-rune text on pages 49 to 51). The trivial case is the rune-prime-values given in the Gematria and the other 58 are their Forward or Reverse position starting from ᚠ=0 for the 1st set, then shifting so we start with ᚢ=0 for the 2nd set, then shifting so we start with ᚦ=0 for the 3rd set etc. We will call these shifts the ordered shifts, or simple shifts. All shifts can be represented by adding 0 to 28 to either the Forward or Reverse position of the rune being considered and can be easily represented by ‘cipher-discs’ such as these (where we can rotate the rune-disc and read off the number from the other discs):  
  
  
  
    [imgur.com/a/XdK82](http://imgur.com/a/XdK82)  
  
General Encryption  
  
  
After we encode a message to numbers it should be encrypted. We apply some function F() to MT, to generate a cipher-text (CT):  
  
  
    CT = F( MT )  
  
  
Similar to encoding our message as numbers there are limitless possibilities for F() and things can get even more complicated: F might be a function that takes more than one variable, we might have to combine MT with a key, K:  
  
  
    CT = F( MT, K )  
  
  
Or F() might actually be two functions, G() and H() with two different keys, K1 and K2:  
  
  
    CT = F( MT, K1, K2 ) = H( G( MT, K1 ) , K2 )  
  
  
and so on. One of our methods for generating rune distributions as observed is basically of this type, where K2 is the previous cipher text rune and H is an addition operator:  
  
  
    [cicada3301.boards.net/thread/30/general-approach-solving-liber-primus](http://cicada3301.boards.net/thread/30/general-approach-solving-liber-primus)  
  
  
The Parameter Space Grows and Grows  
  
  
As an example of how large the parameter space grows we shall count the number of different CT that can be produced under the following conditions:  
  
  
    1. F() will be simple arithmetic operations: Addition, Subtraction, Multiplication, and Division. Page 56 was decrypted with a key of the primes and the addition operator. Tables showing visually how to combine two runes with these operators can be found here:  
  
    [cicada3301.boards.net/thread/31/combining-runes-1-visual-representation](http://cicada3301.boards.net/thread/31/combining-runes-1-visual-representation)  
  
      
    2. The key, K, will be the primes.  
  
  
    3. 4 simple arrangements of the MT and K will be considered: reading left-to-right or reading right-to-left, (many other reasonable transpositions could be applied). Our 2 keys and 2 MT are:  
  
  
        a. MT = F, O, R, A, L, L, I, S, S, A, C, R, E, D  
  
        b. MT\_reverse = D, E, R, C, A, S, S, I, L, L, A, R, O, F  
  
        c. K = 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43  
  
        d. K\_reverse = 43, 41, 37, 31, 29, 23, 19, 17, 13, 11, 7, 5, 3, 2  
  
  
    4. Consider all possible Gematria Positions (obeying the ordering principle)  
  
  
    5. Iterate over all possible ordered shifts. In the following Iterators, indicated by i,j,k will all be assumed to encapsulate a loop where they take the value 0, 1, 2,3,4 etc. This enables us to write compact statements for all the possibilities of applying ordered shifts.  
  
  
    6. As all the pages are expressed in runes each CT must be a list of numbers modulo 29 (% 29).  
  
  
For Addition we have this many different CT:  
  
  
    CT = ( MT + i ) + ( K + j ) + k % 29  
  
  
However, most of the CT are equivalent to another example, (this is easy to see, i=0, j=0 and k=1 will be the same as i=0,j=1,k=0 etc). For this operator there are 116 unique CT.  
  
  
For Subtract there are 290 unique CT from:  
  
  
    CT = ( (MT + i ) - ( K + j ) ) + k % 29 and ( ( K + j ) - ( MT + i ) ) + k % 29  
  
  
We must check the two orders of MT and K subtraction is not associative, ( 5 – 3 != 3 – 5 ).  
  
  
For Multiplication we have 3364 different CT from:  
  
  
    CT = ( (MT + i ) \* ( K + j ) ) + k % 29  
  
  
For Division (which is also not associative) we have 195112 different CT from:  
  
  
    CT = ( (MT + i ) / ( K + j ) ) + k % 29 and ( ( K + j ) / ( MT + i ) ) + k % 29  
  
  
For the simplest operators and the simplest way of encoding the runes and the two simplest transpositions there are 195,348 different CT.  
  
  
Decrypting: the Growth of the Growth of the Parameter Space  
  
  
The above is just for encrypting, when decrypting things grow even more. We don’t know MT and we don’t know K. This means that a “dumb” brute-force approach would have to check every possible word combination and every possible Key. ***We assume we know: the CT rune is in the same position for the MT, the word lengths and sentence lengths.*** As we assume we know the word lengths from the rune spacing that gives us: CT word lengths = 3, 3, 2, 5. A dictionary of rune-words has 85, 885, 6853 words of length 2, 3 and 5 giving a total number of combinations as 456,232,478,625. Given the size of this number it is easy to see why even crossing off Addition and K = primes is not so simple.  
  
  
If we assume that K is not the primes then things become even worse, because then K could be a combination of dictionary words OR any other reasonable integer sequence you can think of, such as the entire On-Line Encyclopedia of Integer Sequences (OEIS) catalog, with millions and millions of numbers, the runes, the first letter from page 17 of the last 50 Wednesday editions of the New York Times… Even checking if a shifted key comes from the prime sequence is not trivial, and then how would you effectively check it’s not ‘every other prime’ or any of the above possibilities?  
  
  
    [cicada3301.boards.net/thread/19/checking-key-sequences-practical-guide](http://cicada3301.boards.net/thread/19/checking-key-sequences-practical-guide)  
  
But we can be smart… Can’t we?  
  
  
Of course we can, and clearly we must be (maybe the smart thing to do is to find the deep-web-page?). We’ve tried to show that many educated guesses and assumptions must be made, however, as soon as you try and be smart you will be cutting the parameter space (something we are tacitly doing all the time). But, by cutting the space you are increasing the chances of a) making a mistake and b) missing the answer - this partly explains why it is nearly impossible to rule out a particular encryption method. We can tell you that we don’t think the pages are encrypted with a simple operator and key from an OEIS sequence or dictionary words – but we can’t prove it.  
  
  
Final Thoughts  
  
  
We have seen how the parameters space just grows and grows with even the most simple of assumptions. The size of the parameter space also makes collaboration more difficult, because the ‘transactional costs’ of collaboration. Clearly, it is theoretically possible to collaborate and narrow the search space by testing ideas, but the practical implications are too great for what are effectively a loose federation of independent solvers. These transactional costs include:  
  
  
    • Potential collaborators are from disparate parts of the world, speaking different languages, both natively and in terms of computer language, with different skills and knowledge.  
  
    • The problem parameter space is so vast that reasonable cuts to the parameter space must be made which means reaching an agreement on what to test, independently searching, or agreeing on methods and cross-checking results.  
  
    • Real life gets in the way  
  
  
It seems that all the above is very deliberate and planned by the puzzle makes: this 2013 message may or may not be genuine, but it seems to contain much truth:  
  
  
    [pastebin.com/VvtLxbsb](http://pastebin.com/VvtLxbsb)  
  
  
**Therefore:**Try things yourself *Believe nothing [from this book] except what you know to be true. Test the Knowledge. Discover your truth….*   
But please, also, come and join us in #cicadasolvers to share your ideas, thoughts and observations, and remember *Patience is aVirtue* the channel is not active every second of every day.