[mortlach](https://cicada3301.boards.net/user/3" \o "@mortlach) Aug 27, 2016 at 6:56pm [a2e7j6ic78h0j](https://cicada3301.boards.net/user/135) likes this

**Introduction**  
  
Often when encrypting / decrypting, you want to combine a message-rune with an encryption key to give a cipher-rune. The encryption key can be expressed as another rune. Therefore, it is informative to investigate possible ways that this can be done. Presented below are some common examples. The runes have two numbers that can represent them, their position (pos) in the Gematria Primus or their rune-prime-equivalent (pri). So there are four possible combinations, (where r is any rune):  
  
r(pos) r(pos), r(pri) r(pri), r(pos) r(pri) , r(pri) r(pos)  
  
All the results below are expressed in a Combination Table with the horizontal axis the first rune and the vertical axis the second rune. Below, the tables are colored and the value in each box is the result of the combination (expressed as an r(pos), but r(pri) or the rune symbol etc. could be used without any loss of generality). Ideally (but not necessarily) each combination will give a reversible result, meaning given the cipher-rune, the key-rune and the operator we can uniquely determine the message-rune. This reversibility can be measured (in some sense) by the number of different results in each row and column of the table, and that number is also given along the bottom and right axes. 100% reversible operations will have all numbers 0 to 28 in each row and column similar to a Sudoku puzzle.  
  
Normally, we prefer 100% reversibility, but because this is a puzzle, because many runes are very uncommon in [Runeglish](http://cicada3301.boards.net/thread/13/runeglish-complete-beginners-introduction)text (e.g. AE) and because life isn’t always perfect it is entirely possible that some of the slightly less 100% reversible schemes could have been used. For example adding rune-prime-equivalents doesn’t give 100% reversibility, but it probably does have enough reversibility for an intelligent solver to fill in the gaps. This irreversibility happens because we work with modular arithmetic. We want to represent the result as one of only 29 runes, therefore we must take the result mod 29 ( % 29 ).  
 **Addition and Subtraction**  
  
[A simple method of combination is addition/subtraction. For positions this gives the vigenere table. Subtraction can be seen as the inverse of addition (and the associativity of the operators can also be seen).](http://imgur.com/a/5BfOf)  
  
The tables can also show how the runes “shift.” Shifting of the rune positional index has happened in previous puzzles. Basically, shifts rotate the columns, rows or columns and rows along the table, when a row gets to the end it cycles back to the start again. This happens because we are applying our operators using modular arithmetic. [This animation shows the addition operator shifting the tables by different values.](http://imgur.com/a/8Hsay)  
  
**Multiplication and Division**  
  
Multiplication is fairly easy and shouldn’t need further explanation. [The animation showing the shifts is nice – and it also shows some of the many different ways shifts can be applied](http://imgur.com/a/8Hsay).  
  
Division mod 29 is a little more complicated. We define division as the solution to:  
  
solution = (a/b) % 29 = a % 29 ( b^-1 % 29 )  
  
I call b^-1 the multiplicative inverse, [it can be found from the multiplication table, using digital roots, or in other ways](http://imgur.com/a/BgD7T).  
  
**Exponentiation, XOR, and others?**  
  
[For fun here are some other ways to combine the runes](http://imgur.com/a/AENeJ). Notice how we are starting to loose a lot of reversibility. There are many other possible schemes, and actually, you don’t need a function, you could create a home-made operator by arranging the runes on the table any way you feel, by hand.  
  
  
\*Comments, questions, suggestions, omissions etc ? please try [#cicadasolvers](http://webchat.freenode.net/?channels=cicadasolvers)  
  
MSGA