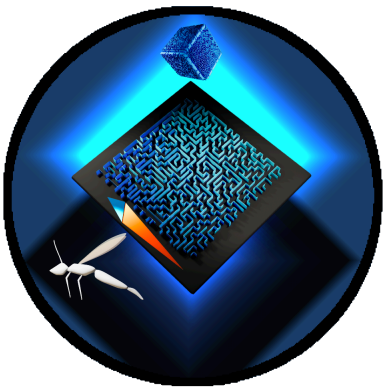
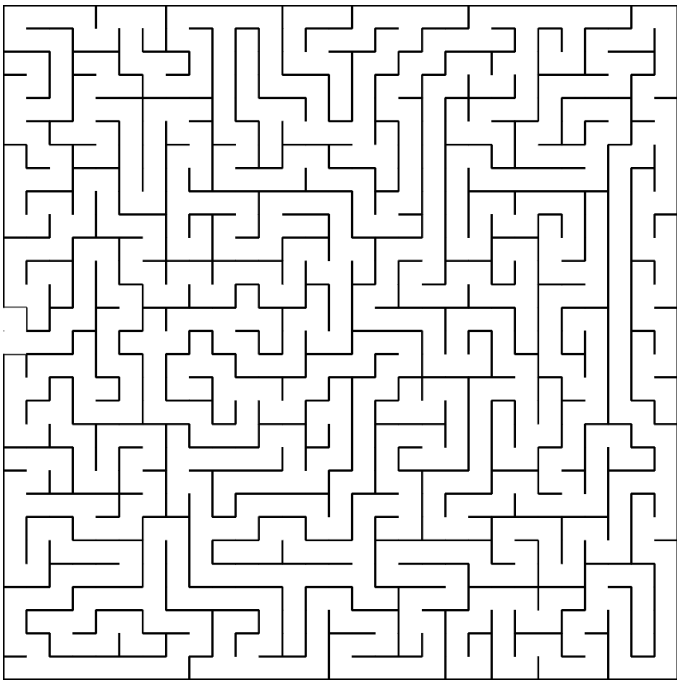


Lead Edge Cryptography

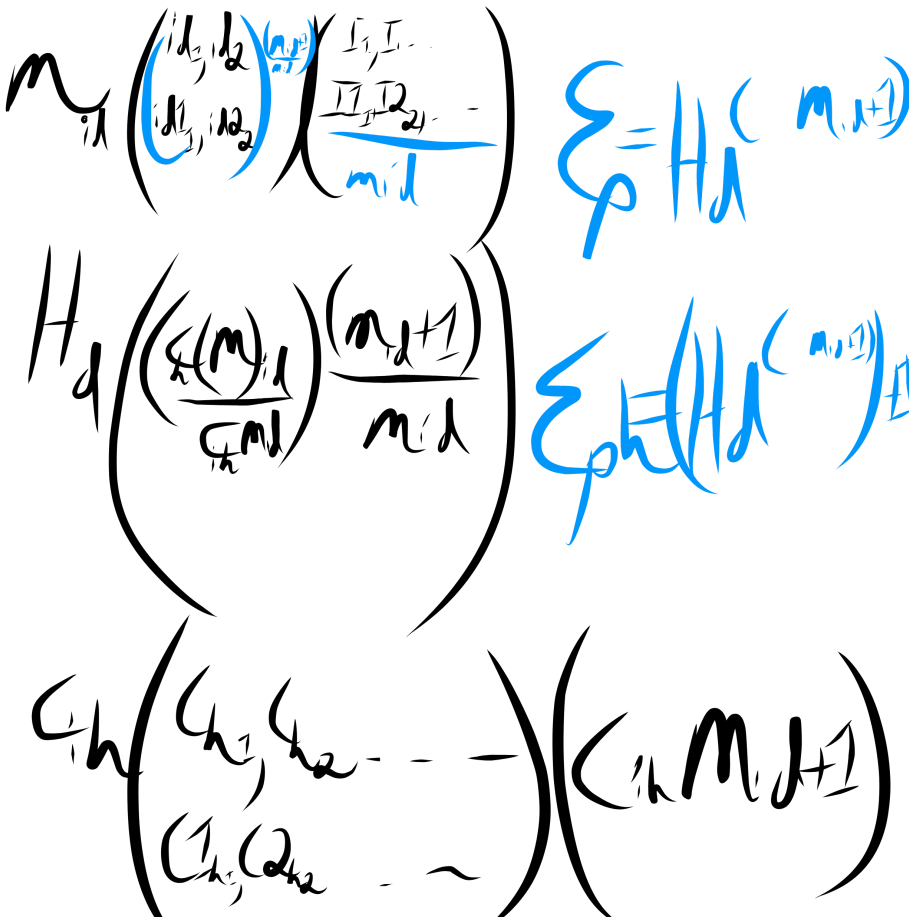
(v.1.0)

Algorithm Legend

M = Maze
id = interchange dimension of the maze points
I = Interchange of Mazes
H = Hashing of the Content by Maze interchange dimension
d = dimension of content to apply Maze interchange dimension
C = Content
ih = index hashing of content
E = Edge Key
p = private Edge Key
pb = public Edge Key
{} = The overall Content or Matrix of Content



Encryption Algorithm



Where Maze interchange dimension points are allocated against themselves to the power of Maze interchange dimension plus 1 then divided by themselves. Further the Maze interchange dimension is multiplied to any inclusion of additional Mazes for interchange on the condition they are divided by the initial Maze interchange dimension:

Mid = { (id1,id2)^((Mid+1)/Mid) }{ (I1,I2,I3)/Mid }

With the Maze interchange dimension set the Maze interchange dimension and Content can be hashed together:

Hd = { ((Ch(M)id)/Cih(Mid))((Mid+1)/Mid) }

Here The direct Content is hashed against the initial Maze before the initial Maze is altered by interchange dimension then they both hashed against the Maze interchange dimension while divided by the overall Content index hash that is multiplied against the Maze interchange dimension. With these stages set the Hash dimension continues to the power of Maze interchange dimension plus 1 then divided by the Maze interchange dimension.

The overall direct Content index hash consistent of any content to be hashed while multiplied against itself of

being multiplied by the Maze interchange dimension plus 1:

Cih = { Cih }{ Cih(Mid}+1 }

These stages are taken over to the private and public Edge Key generators Hash dimension to the power of Content interchange dimension multiplied by Maze interchange dimension plus 1 and Hash dimension to the power of Content interchange dimension multiplied by Maze interchange dimension plus 1 then plus 1 respectively:

Ep = Hd^(CidMid+1)

Epb = (Hd^(CidMid+1))+1

The Algorithm is designed to keep the hashing ability dynamically buildable while allocating dynamic crunching for any potential changes to the interchange network or root network. The Hashing becomes an increase into its Encrypted output whether this be a key or what else may be desired while the Maze interchange dimension output is an increase by the Maze interchange dimension itself again as a crunch into itself.

The only concern for Quantum Superposition Encryption or decryption should only be taken into account regarding the surgical Maze Property of only having 2 openings, no more, no less thus should abide by care where Superposition is defined by (b+b)*(a^2)/r for non conductive superposition and (b+b)*(p(a^2))/r for conductive superposition.

Quantum Superposition is set by b the Element then multiplied by its proportional liquid foundation that is either multiplied by its proportional gas state or not then divided by the overall new resource of itself.

Therefore when using the Lead Edge Cryptography for Encryption or Decryption you can be safely minimal taking the Maze into account by how you present the methods or layers to the methods as well as providing the keys. You can also provide the Content that is Encrypted in a digital Know form where the result it's the original content exposed by its Encryption but with the resulting natural Sigma of {Know} being appended to the original data after Encryption because the Maze is at root to the Cryptography meaning the Encryption finds the main context of the data at its sigma truth, down to the same color using visual data as the subject for example, during Encryption then formulates the interchange dimension Cryptography for regenerating this same data as its Sigma

Encryption abiding by the initial Maze used for Cryptography. This is really only necessary for sensor type Cryptography where someone might see visual data out of their peripheral vision or directly and know everything about what they wanted to do with the data that is not for them but would have to do so by the new exposure which would force their actions to leave behind that natural {Know}. Further when deciding on utilizing this technique it can be found only statically necessary in the long term, say 200 years after the origins are no longer existent by example.

With this said you could use the digital {Know} not as {Know} or even as {Know} directly but use its logic and further not even expose the DATA by a means of Encryption if you choose to Safeguard against Quantum Superposition or Similarities. Effectively using Lead Edge Cryptography you will not need to use the Encryption byproduct result {Know} Logic at all. {Know} occurs because a true Maze is surgical, 2 openings, no more, no less or a goal of 2 locations, no more, no less as well no islands of space or wall segments.

Decryption Algorithm

For Decryption you may just reverse Ep and Epb at the square root of if you choose to use the Algebraic style in your method:

Decryption Ep by memory logic methods = √Hd^(Mid-1)

Decryption Ep directly = √Hd^(Mid+1)

Decryption Epb by memory logic methods = √(Hd^(Mid-1))-1

Decryption Epb directly = √(Hd^(Mid+1))+1