difficulty= 2^(256-x)/target。
在这里假定没有leading zero x=0。这样difficulty的含义就是需要计算的哈希次数。
比如difficulty=2^40，那么就是要哈希出40个0来，或者说单次哈希成功的概率
p=1/difficulty=2^-40 (1)

那么第n次才恰好哈希到40个0的概率就是p*（1-p）^(n-1) ，前n-1次都失败，最后一次成功。
可以暂时称这样的哈希为出块哈希。

给定difficulty，前n次能找到出块哈希的概率，也叫概率累计函数P(n)：

$$P(n)=\sum_{k=1}^{n} p*(1-p)^{n-1}=p*\frac{1-(1-p)^{n}}{1-(1-p)}=1-(1-p)^{n}$$

P(n)在0到1之间，当n->inf的时候，P(n)=1。

上面是推导，下面是正题：
模拟恰好第N_hsr次哈希出，步骤如下：
1．先找出一个0-1均匀分布的随机数rand
2．设立不等式P(n-1)<rand<P(n)
3．求出N_hsr。
求解可得：

$$N\_hsr=ceil(\frac{\log(1-rand)}{\log(1-p)}) \quad (2)$$

其中ceil表示向上取整。

不妨设当前难度为D时，某个用户的算力HRworker=D/120，则该用户可以在当前难度下平均T=120秒出块，利用随机数rand和p=1/D计算出此次出块需要的哈希次数N_hsr之后，可以算出该用户此次出块所耗时间：
solvetime=N_hsr/HRworker
其中N_hsr由式子(2)求出。由此就解决了模拟solvetime的问题。

类似，假定攻击者算力是诚实矿工的m倍：
HRAttacker=m*HRworker
那么，在被攻击期间当难度是D的时候，该次出块的时间就是N_hsr/HRworker/(m+1)。