CryptoHello 工作量证明方案

本项目的目标是设计一种在 CPU 上可以较为高效工作，但是在 GPU、FPGA、ASIC 上却难以高效事先的工作量证明方案。其基本描述是设计一个单向函数 H，满足 $y=H(x)$。H 应满足的要求类似于散列函数，即单向性、雪崩性、随机性，其计算效率在 CPU 上较高，但是在 GPU、FPGA、ASIC 上难以发挥出性能。

一、总体技术方案

1.1 基本原则

为了防止 GPU、FPGA、ASIC 上的高效实现，H 的设计需要满足以下特征：

（1）内存受限。需要较大空间的内存容量，而且对内存有大量的随机访问。考虑到 CPU 和 GPU 的 cache 容量差异，内存容量大约在 2MB 左右。

（2）高度串行化。H 的计算过程中基本没有可以并行执行的成分，必须严格按照串行执行；

（3）大量的控制相关性。即 H 中需要执行比较多的条件分支指令，而且这些分支转移的概率均接近 50%；

（4）多种负载不均衡的散列函数组合。这主要是防止 ASIC 芯片中针对每种散列函数专门定制设计。如果是有很多种散列函数，将大幅度增加 ASIC 芯片的芯片面积；

（5）可以充分利用 CPU 中的 SIMD 指令，加速计算 CPU 上的计算过程。

1.2 总体技术方案

【定义 1】 散列函数族 h_i，$0 \le i \le 15$，共计 16 种不同的散列函数。每个散列函数的输出固定为 5 个 32 位字。

【定义 2】工作存储器 M，总容量为$|M|$字节，$|M|$能被 16 整除。

【定义 3】伪随机函数发生器 *seed*(uint32 s)为设置随机数发生器种子，*rand*()为随机数发生器结果，返回 32 位无符号整数。

【定义 4】单向函数 $y=H(x)$，其中 y 为 5 个 32 位字。

【定义 5】reduce_bit(x, y)将 x 的内容规约到 y 位。

算法分为三步骤：1）初始化工作存储器 M；2）修改工作存储器 M 内容；3）根据工作存储器内容产生最后的结果。

（1）初始化工作存储器

按照以下方式初始化工作存储器 M。

输入：x

输出：M

中间变量：a, b 均为长度为 5 个字的向量，a[c1:c2]表示其中的第 c1:c2 个字节

1. $a=h_0(x)$

2. i 从 0 到$|M|$/16-1 循环　　//$|M|$=2MB 时，循环 128K 次

2.1　$M[16*i:16*i+15]=a[0:15]$;
2.2　$t=reduce\_bit(a[16:19],4)$;
2.3　$a=h_t(a[0:15]|i)$;

（2）修改工作存储器内容

输入：M　$M[addr]$表示地址为 $addr$ 的一个字节

输出：*M*

1. $a=h_0$(*M* 的逆序)；*r*=0; //必须完成整个存储器的初始化后才能进行

2. *i* 从 0 到 *C**(|*M*|/16-1)循环 //循环 *C* 次，共计 2*C*|*M*|次随机存储器读写

2.1 *seed*(*a*[16:19]); //初始化随机数发生器

2.2 *j* 从 0 到 15 循环 //修改存储器内容，并产生新的散列函数输入

2.2.1 *addr*=*rand*() mod |*M*|+*reduce_bit*(*r*,5)<< *reduce_bit*(*r*,4); //产生随机地址

2.2.2 *t*=*M*[*addr*];

2.2.3 *M*[|*M*|-*addr*]= *t* XOR *a*[*i*]; //修改工作存储器

2.2.4 *r*=*a*[*i*]=*t*; //修改 a

2.3 *t*=*reduce_bit*(*a*[16:19],4);

2.4 对 *a* 构成的字节数组按照从小到大排序;

2.5 $a=h_t$(a[0:15]|*i*);

简要说明：

2 步中的 *C* 是一个可以调节参数，用于调整系统的执行速度。

2.2.1 步使得每次存储器访问的地址随机，而且地址依赖于伪随机序列和上次访存的内容。

2.2.3 步的写入地址与 2.2.2 步的读出地址刚好对称，使得存储器地址的局部性进一步减少。

2.4 步的字节排序将增大 GPU 执行的难度，但是在 CPU 上可以使用 SIMD 指令并行加速。

（3）根据工作存储器内容产生最后结果

输入：*M*

输出：*y*

1. $y=h_0$(*M*[0:16]|0);

2. *i* 从 1 到|*M*|/16-1 循环

2.1 *t*= *reduce_bit*(*y*[16:19],4);

2.2 *y*[0:15]=*y*[0:15] XOR *M*[16**i*, 16**i*+15];

2.3 $y=h_t(y[0:15]|i)$;