零知识证明与zk-SNARK

星辰实验室

钟林

lynndell2010@gmail.com

预备知识

多项式时间算法：能够快速计算出结果的算法。

① 已知私钥sk和椭圆曲线生成元G，能够快速计算出公钥PK:=sk•G 。

② 已知原象x和哈希函数SHA256，能够快速计算出函数值Y:=SHA256(x)。

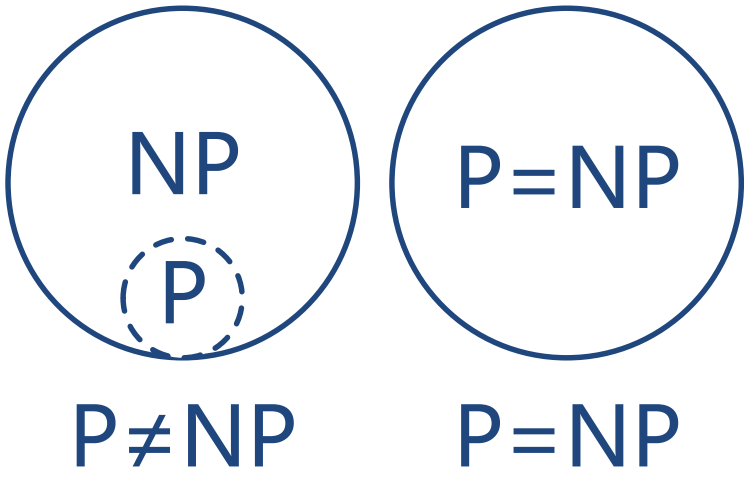
非多项式时间算法：包括指数时间算法、亚指数时间算法。

以下典型算法需要指数时间：

① 已知公钥PK和椭圆曲线生成元G，不能在多项式时间内计算出私钥sk，而需要指数时间才能计算出私钥sk，使得PK=sk•G成立。

② 已知函数值Y和哈希函数SHA256，不能在多项式时间内计算出原象x，而需要指数时间才能计算出原象x，使得Y=SHA256(x)成立。

预备知识



P问题≠NP问题

预备知识

P问题：在多项式时间内可计算的问题。以下问题是典型的P问题

① 已知原象x和哈希函数SHA256，能在多项式时间内计算出函数值Y，使得Y=SHA256(x)成立。
② 已知私钥sk和椭圆曲线生成元G，能在多项式时间内计算出公钥PK，使得PK=sk•G成立。

NP问题：
（1）多项式时间内不可计算的问题，需要指数时间或亚指数时间；
（2）但是，一旦已知解，则能够在多项式时间内验证解是否正确。以下问题是典型NP问题

① 已知函数值Y和哈希函数SHA256，不能在多项式时间内计算出原象x，使得Y=SHA256(x)成立。但是，一旦已知原象x，则能够在多项式时间内验证x是否正确。
② 已知公钥PK和椭圆曲线生成元G，不能在多项式时间内计算出私钥sk，使得PK=sk•G成立。但是，一旦已知私钥sk，则能够在多项式时间内验证sk是否正确。

NP问题的本质是单向性，不能快速逆向求解，但是能够快速正向验证！

预备知识：重要的NP问题：多项式整除

星辰实验室

已知阶为n的多项式z(x)、阶小于或等于n的三组多项式 $u_0(x),\dots u_n(x);v_0(x),\dots v_n(x);w_0(x),\dots w_n(x)$，这些多项式等于零的解相同。不能在多项式时间内计算出向量 $s=(1,s_1,\dots,s_m)$ ，满足以下**整除关系**

$$z(x)\left|\left(\sum_{i=0}^{m}s_i\cdot u_i(x)\right)\cdot\left(\sum_{i=0}^{m}s_i\cdot v_i(x)\right)-\left(\sum_{i=0}^{m}s_i\cdot w_i(x)\right)\right.$$

原理分析：

① 如果向量元素s_i的取值空间为a，则将公式 $\left(\sum_{i=0}^{m}s_i\cdot u_i(x)\right)\cdot\left(\sum_{i=0}^{m}s_i\cdot v_i(x)\right)-\left(\sum_{i=0}^{m}s_i\cdot w_i(x)\right)$ 称为二次算法多项式，简称为QAP多项式。如果s_i的取值空间为0/1，则称为二次扩张多项式，简称为QSP多项式。因此，QAP/QSP多项式的构造空间分别为$a^m/2^m$，呈**指数空间**。

② z(x)=0有n个解，QAP/QSP多项式等于零的解数量范围为(n,2n]。所以，除z(x)=0的n个解以外，QAP/QSP多项式等于零还有其他解。因此，可以将QAP/QSP多项式除以z(x)得到商多项式h(x)。h(x)本质上就是QAP/QSP多项式等于零的其他解构成的多项式。

③ 如果不知道向量s，则只能随机选择一个向量s，计算QAP/QSP多项式，然后检测z(x)与之是否满足整除关系。如果满足，则接受，否则拒绝。因此，需要**指数时间**才能够暴力搜索出向量s。但是，一旦给定向量s，则能够快速验证构造出的QAP/QSP多项式是否与z(x)满足整除关系。

因此，多项式z(x)与QAP/QSP多项式的整除关系，满足单项性，构成NP问题！

预备知识

密码学的本质：

① 将数据放到椭圆曲线离散对数点上，形成离散对数困难。

② 椭圆曲线离散对数点是群元素，群元素可进行**二元运算。**

零知识证明与zk-SNARK核心技术：

由于用户隐私保护、数据保密性、区块链扩容等应用需求，

- **证明方**：将上述NP问题中QAP/QSP多项式、商多项式和z(x)多项式这三个多项式的**系数**放到椭圆曲线离散对数点上（专业术语称为：**多项式承诺**），形成离散对数困难。
- **验证方**：对生成的椭圆曲线离散对数点进行**二元运算，重构整除关系**，能够快速验证向量s的正确性，却不知道向量s。

预备知识

命题：多项式的值表达等价于多项式的系数表达

① 已知变量和多项式的值$(x_1,f_1),\ldots,(x_n,f_n)$，能够在多项式时间内求解多项式的系数$k_1,\ldots,k_n$。设多项式为

$$f(x)=k_1\cdot x^1+k_2\cdot x^2+\ldots+k_n\cdot x^n$$

因此，能够将方程中的系数$k_1,\ldots,k_n$快速求解出来。
其他快速求解方法有：**拉格朗日插值法**、快速傅里叶变换等。
为计算方便，后续的举例将使用**拉格朗日插值法**求解多项式的系数。

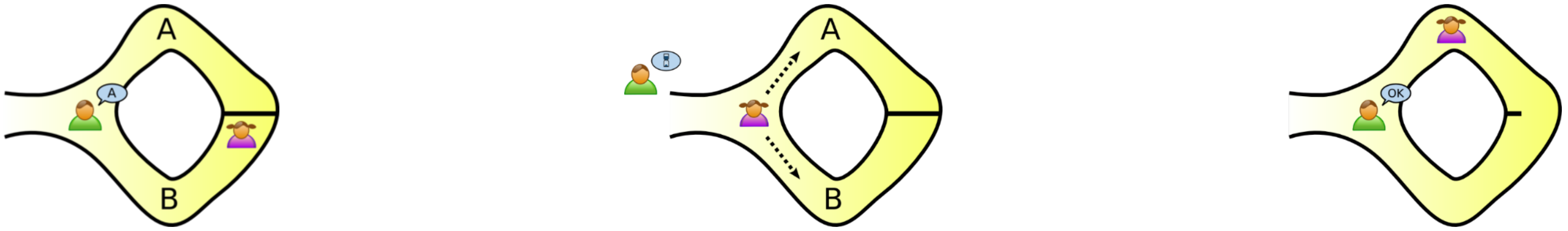
② 已知变量$x_1,\ldots,x_n$和多项式的系数$k_1,\ldots,k_n$，能够在多项式时间内计算多项式的值$f_1,\ldots,f_n$

零知识证明概念

零知识证明：
证明方先在不揭露秘密的情况下，使验证方相信某个论断是正确的。

例1：小明在不揭露私钥的情况下，把创世块上的比特币支付到其预先声明的区块，则验证方相信小明是中本聪。

例2：



① 小红向小明展示洞穴里有个门

② 小红向小明证明其知道开门秘诀，但是不能泄露开门秘诀

① 小明去外面（看不见小红）

② 小红选择通道A或B藏起来

① 小明随机选择通道A或B，**要求小红从规定的通道中出来**

② 小红确实从规定的通道中出来

核心思想

如果图2和图3的游戏仅运行几次，小红可能是碰巧藏在规定的通道中，则能够从规定的通道中走出来，她是不知道开门秘诀的！但是，如果游戏重复多次，则小红碰巧成功的概率呈指数降低$1/2^n$。只有知道开门秘诀才可以每次都从规定的通道中走出来；因此，如果小红每次都正确，则小明肯定认可小红知道开门秘诀，但是开门秘诀没泄露。

Sigma零知识证明协议

预备知识	① 公钥与私钥满足**椭圆曲线离散对数关系：PK=sk•G**
	② 椭圆曲线离散对数困难问题：已知公钥PK和G，无法在多项式时间内计算私钥sk。

Sigma零知识证明的目标：证明方证明其知道ω且ω满足离散对数关系Q=ω•G

Sigma协议包括以下5个步：

① **系统参数**：椭圆曲线生成元：**G**

② **承诺**：证明方选择一个随机数r，计算并发送椭圆曲线离散对数点：**C:=r•G**

③ **挑战**：验证方选择一个随机数**e**并发送

④ **响应**：证明方基于随机数e构造并发送**线性关系：z:=r+e•ω**

⑤ **验证**：验证方首先，基于e和z分别计算椭圆曲线离散对数点：**z•G, e•Q**

然后，在椭圆曲线离散对数点上重构线性关系：**z•G=C+e•Q**

如果**线性关系**重构成功，则接受，否则拒绝

公式展开：左边：**z•G=(r+e•ω)•G**，右边：**C+e•Q=r•G+e•ω•G=(r+e•ω)•G**，**等式成立，验证成功**

系统参数：G（双方均知道）；秘密知识：r, ω（仅证明方知道）
公开参数：Q, C, e, z,（双方均知道）；z•G, e•Q（双方均可计算出）

零知识证明**扩展**

Sigma协议只证明：

ω满足离散对数关系Q=ω•G

Sigma协议扩展：zk-SNARK简洁非交互式零知识论证：

ω满足任意多项式时间运算关系Y=F(ω)

Sigma协议中的离散对数关系Q=ω•G是天然的NP问题。但是，zk-SNARK协议中的任意多项式时间运算关系Y=F(ω)不是天然的NP问题。

接下来使用zk-SNARK中的7个等价转化关系：基于Y=F(ω)构造多项式整除关系，满足NP困难，并使用椭圆曲线离散对数困难，构造零知识。

zk-SNARK 7个等价转化关系如下：

ω 满足任意多项式时间运算关系 Y=F(ω)	⟺	ω 满足 R1CS约束 (电路约束)	⟺	向量s （内积） 多维向量 (u_i,v_i,w_i)	⟺	向量s （内积） 矩阵 U,V,W	⟺	向量s （组合运算） 多项式 U(x),V(x),W(x)	⟺	目标多项式 （整除） QAP多项式 **（NP问题）**	⟺	QAP多项式 目标多项式 商多项式 （多指数运算） **（证据）**	⟺	椭圆曲线 离散对数 （双线性映射） 双线性群

zk-SNARK协议

ω 满足任意多项式时间运算关系 Y=F(ω)	⟺	ω 满足R1CS约束(电路约束)	⟺	向量s（内积）多维向量 (u_i,v_i,w_i)	⟺	向量s（内积）矩阵 U,V,W	⟺	向量s（组合运算）多项式 U(x),V(x),W(x)	⟺	目标多项式（整除）QAP多项式（NP问题）	⟺	QAP多项式 目标多项式 商多项式（多指数运算）（证据）	⟺	椭圆曲线 离散对数（双线性映射）双线性群

$$a \in \{0,1\} \quad \Updownarrow \quad \{(1-a)\times a = 0\}$$

$$a \oplus b = c \quad \Updownarrow \quad \begin{Bmatrix} (1-a)\times a = 0 \\ (1-b)\times b = 0 \\ (1-c)\times c = 0 \\ (2a)\times(b) = (a+b-c) \end{Bmatrix}$$

$$x^4 + x^3 + x^2 + x = 120 \quad \Updownarrow \quad \begin{Bmatrix} s_1 = x * x \\ s_2 = s_1 * x \\ s_3 = s_2 * x \\ s_4 = s_1 + x \\ s_5 = s_4 + s_2 \\ 120 = s_5 + s_3 \end{Bmatrix}$$

这些阶为1的等式限定了算法的运算规则，能够用电路约束表达同样的运算规则。电路约束即能用硬件电路或FPGA实现，也可用软件实现等价的功能。因此，阶为1的等式等价于电路约束。

阶为1的等式(R1CS约束)=电路约束

阶为1的乘法等式=乘法约束

阶为1的加法等式=加法约束

关键结论：任意多项式时间算法均可以拆为阶为1的等式，等价于电路约束

zk-SNARK协议

ω 满足任意多项式时间运算关系 Y=F(ω) ⟺ ω 满足 R1CS约束 (电路约束) ⟺ 向量s（内积）多维向量 (u_i,v_i,w_i) ⟺ 向量s（内积）矩阵 U,V,W ⟺ 向量s（组合运算）多项式 U(x),V(x),W(x) ⟺ 目标多项式（整除）QAP多项式（**NP问题**） ⟺ QAP多项式 目标多项式 商多项式（多指数运算）（**证据**） ⟺ 椭圆曲线 离散对数（双线性映射）双线性群

阶为1的等式=电路约束

阶为1的乘法等式=乘法约束

阶为1的加法等式=加法约束

算法的R1CS约束优化：可将加法约束耦合到乘法约束中

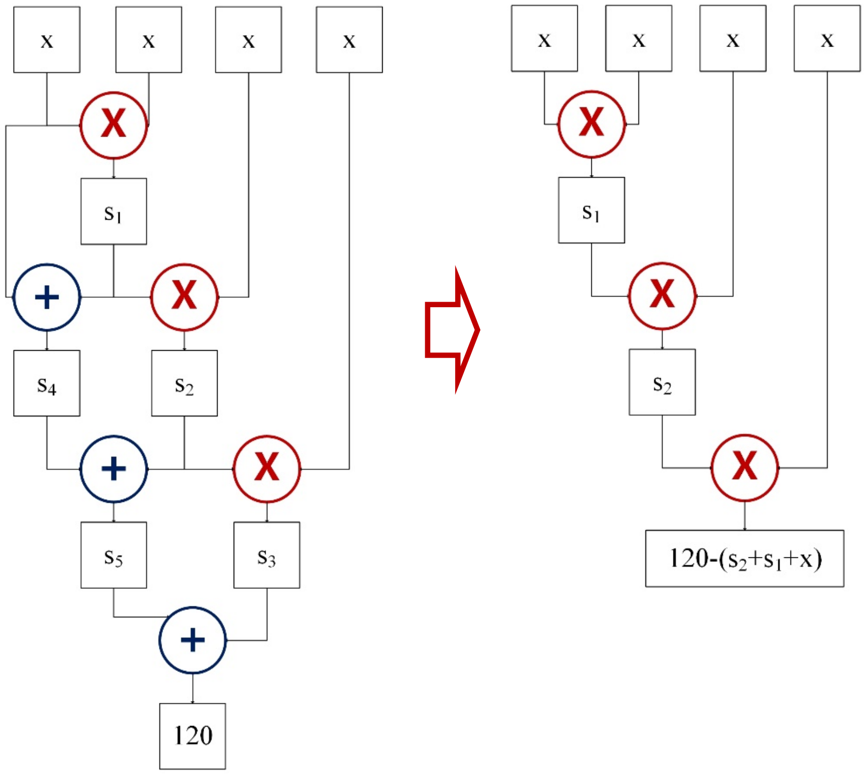
$$x^4+x^3+x^2+x=120$$

$$\Updownarrow$$

$$\begin{cases} s_1 = x*x \\ s_2 = s_1*x \\ s_3 = s_2*x \\ s_4 = s_1+x \\ s_5 = s_4+s_2 \\ 120 = s_5+s_3 \end{cases}$$

$$\Updownarrow$$

$$\begin{cases} s_1 = x*x \\ s_2 = s_1*x \\ 120-x-s_1-s_2 = s_2*x \end{cases}$$



zk-SNARK协议

ω 满足任意多项式时间运算关系 Y=F(ω)	⟺	ω 满足R1CS约束(电路约束)	⟺	向量s（内积）多维向量(u_i,v_i,w_i)	⟺	向量s（内积）矩阵U,V,W	⟺	向量s（组合运算）多项式U(x),V(x),W(x)	⟺	目标多项式（整除）QAP多项式（NP问题）	⟺	QAP多项式目标多项式商多项式（多指数运算）（证据）	⟺	椭圆曲线离散对数（双线性映射）双线性群

$$s=(statement, witness)=(1,out,x,s_1,s_2)$$

$$\{s_1=x*x\}\Leftrightarrow\begin{cases}\vec{s}\cdot[0,0,0,1,0]=[1,out,x,s_1,s_2]\cdot[0,0,0,1,0]=s_1\\ \vec{s}\cdot[0,0,1,0,0]=[1,out,x,s_1,s_2]\cdot[0,0,1,0,0]=x\\ \vec{s}\cdot[0,0,1,0,0]=[1,out,x,s_1,s_2]\cdot[0,0,1,0,0]=x\end{cases}$$

$$\{s_2=s_1*x\}\Leftrightarrow\begin{cases}\vec{s}\cdot[0,0,0,0,1]=[1,out,x,s_1,s_2]\cdot[0,0,0,0,1]=s_2\\ \vec{s}\cdot[0,0,0,1,0]=[1,out,x,s_1,s_2]\cdot[0,0,0,1,0]=s_1\\ \vec{s}\cdot[0,0,1,0,0]=[1,out,x,s_1,s_2]\cdot[0,0,1,0,0]=x\end{cases}$$

$$\{120-x-s_1-s_2=s_2*x\}\Leftrightarrow\begin{cases}\vec{s}\cdot[0,1,-1,-1,-1]=[1,out,x,s_1,s_2]\cdot[0,1,-1,-1,-1]=120-(x+s_1+s_2)\\ \vec{s}\cdot[0,0,0,0,1]=[1,out,x,s_1,s_2]\cdot[0,0,0,0,1]=s_2\\ \vec{s}\cdot[0,0,1,0,0]=[1,out,x,s_1,s_2]\cdot[0,0,1,0,0]=x\end{cases}$$

$$s_1=x*x$$
$$s_2=s_1*x$$
$$120-x-s_1-s_2=s_2*x$$
$$\Updownarrow$$
$$s(w_1(x),\ldots,w_3(x))=s(u_1(x),\ldots,u_3(x))*s(v_1(x),\ldots,v_3(x))$$

关键结论：ω满足R1CS约束等价转化为向量与多维向量的内积

zk-SNARK协议

ω 满足任意多项式时间运算关系 Y=F(ω) ⟺ ω 满足 R1CS约束 (电路约束) ⟺ 向量s （内积） 多维向量 (u_i,v_i,w_i) ⟺ 向量s （内积） 矩阵 U,V,W ⟺ 向量s （组合运算） 多项式 U(x),V(x),W(x) ⟺ 目标多项式 （整除） QAP多项式 （**NP问题**） ⟺ QAP多项式 目标多项式 商多项式 （多指数运算） （**证据**） ⟺ 椭圆曲线 离散对数 （双线性映射） 双线性群

$$\{s_1 = x*x\} \Leftrightarrow \left\{\begin{array}{l} \vec{s}\cdot[0,0,0,1,0]=[1,out,x,s_1,s_2]\cdot[0,0,0,1,0]=s_1 \\ \vec{s}\cdot[0,0,1,0,0]=[1,out,x,s_1,s_2]\cdot[0,0,1,0,0]=x \\ \vec{s}\cdot[0,0,1,0,0]=[1,out,x,s_1,s_2]\cdot[0,0,1,0,0]=x \end{array}\right\}$$

$$W=\begin{bmatrix} 0,0, & 0, & 1, & 0, \\ 0,0, & 0, & 0, & 1, \\ 0,1, & -1, & -1, & -1, \end{bmatrix}$$

$$\{s_2 = s_1*x\} \Leftrightarrow \left\{\begin{array}{l} \vec{s}\cdot[0,0,0,0,1]=[1,out,x,s_1,s_2]\cdot[0,0,0,0,1]=s_2 \\ \vec{s}\cdot[0,0,0,1,0]=[1,out,x,s_1,s_2]\cdot[0,0,0,1,0]=s_1 \\ \vec{s}\cdot[0,0,1,0,0]=[1,out,x,s_1,s_2]\cdot[0,0,1,0,0]=x \end{array}\right\}$$

$$U=\begin{bmatrix} 0,0,1,0,0 \\ 0,0,0,1,0 \\ 0,0,0,0,1 \end{bmatrix}$$

$$\{120-x-s_1-s_2 = s_2*x\} \Leftrightarrow \left\{\begin{array}{l} \vec{s}\cdot[0,1,-1,-1,-1]=[1,out,x,s_1,s_2]\cdot[0,1,-1,-1,-1]=120-(s_2+s_1+x) \\ \vec{s}\cdot[0,0,0,0,1]=[1,out,x,s_1,s_2]\cdot[0,0,0,0,1]=s_2 \\ \vec{s}\cdot[0,0,1,0,0]=[1,out,x,s_1,s_2]\cdot[0,0,1,0,0]=x \end{array}\right\}$$

$$V=\begin{bmatrix} 0,0,1,0,0 \\ 0,0,1,0,0 \\ 0,0,1,0,0 \end{bmatrix}$$

$$s(w_1(x),...,w_3(x)) = s(u_1(x),...,u_3(x)) * s(v_1(x),...,v_3(x))$$

$$\Updownarrow$$

$$s\cdot W = s\cdot U * s\cdot V$$

关键结论：向量与多维向量的内积等价转换为向量与矩阵的内积

zk-SNARK协议

ω 满足任意多项式时间运算关系 Y=F(ω)	⟺	ω 满足 R1CS约束 (电路约束)	⟺	向量s （内积） 多维向量 (u_i,v_i,w_i)	⟺	向量s （内积） 矩阵 U,V,W	⟺	向量s （组合运算） 多项式 U(x),V(x),W(x)	⟺	目标多项式 （整除） QAP多项式 （**NP问题**）	⟺	QAP多项式 目标多项式 商多项式 （多指数运算） （**证据**）	⟺	椭圆曲线 离散对数 （双线性映射） 双线性群

$$W=\begin{bmatrix}0,0,&0,&1,&0,\\0,0,&0,&0,&1,\\0,1,&-1,&-1,&-1,\end{bmatrix},U=\begin{bmatrix}0,0,1,0,0\\0,0,0,1,0\\0,0,0,0,1\end{bmatrix},V=\begin{bmatrix}0,0,1,0,0\\0,0,1,0,0\\0,0,1,0,0\end{bmatrix}$$

拉格朗日插值多项式

$$f(x)=\sum_{k=1}^{t}f_k\prod_{j=1,j\neq k}^{t}\frac{x-x_j}{x_k-x_j}$$

$$w_1(x)=0\frac{(x-2)(x-3)}{(1-2)(1-3)}+0\frac{(x-1)(x-3)}{(2-1)(2-3)}+0\frac{(x-1)(x-2)}{(3-1)(3-2)}=0$$

$$w_2(x)=0\frac{(x-2)(x-3)}{(1-2)(1-3)}+0\frac{(x-1)(x-3)}{(2-1)(2-3)}+1\frac{(x-1)(x-2)}{(3-1)(3-2)}=\frac{1}{2}(x^2-3x+2)$$

$$w_3(x)=0\frac{(x-2)(x-3)}{(1-2)(1-3)}+0\frac{(x-1)(x-3)}{(2-1)(2-3)}-1\frac{(x-1)(x-2)}{(3-1)(3-2)}=-\frac{1}{2}(x^2-3x+2)$$

$$w_4(x)=1\frac{(x-2)(x-3)}{(1-2)(1-3)}+0\frac{(x-1)(x-3)}{(2-1)(2-3)}-1\frac{(x-1)(x-2)}{(3-1)(3-2)}=-x^2+4x-4$$

$$w_5(x)=0\frac{(x-2)(x-3)}{(1-2)(1-3)}+1\frac{(x-1)(x-3)}{(2-1)(2-3)}-1\frac{(x-1)(x-2)}{(3-1)(3-2)}=-\frac{1}{2}(3x^2-11x+8)$$

$$W(x)=[w_1(x),w_2(x),w_3(x),w_4(x),w_5(x)],U(x)=[u_1(x),u_2(x),u_3(x),u_4(x),u_5(x)],V(x)=[v_1(x),v_2(x),v_3(x),v_4(x),v_5(x)]$$

$$s\cdot W=s\cdot U*s\cdot V\Leftrightarrow s\cdot W(x)-s\cdot U(x)*s\cdot V(x)=0$$

关键结论：向量与矩阵的内积等价于向量对多项式的组合运算

zk-SNARK协议

ω 满足任意 多项式时间 运算关系 Y=F(ω)	⟺	ω 满足 R1CS约束 (电路约束)	⟺	向量s （内积） 多维向量 (u_i,v_i,w_i)	⟺	向量s （内积） 矩阵 U,V,W	⟺	向量s （组合运算） 多项式 U(x),V(x),W(x)	⟺	目标多项式 （整除） QAP多项式 **（NP问题）**	⟺	QAP多项式 目标多项式 商多项式 （多指数运算） **（证据）**	⟺	椭圆曲线 离散对数 （双线性映射） 双线性群

拉格朗日插值多项式的横坐标x=1,2,3，所以引入目标多项式z(x)=(x-1)(x-2)(x-3)

当x=1,2,3， $z(x)=0$ $\quad \mathrm{QAP}:\left(s\cdot W(x)-s\cdot U(x)*s\cdot V(x)\right)=0$

可以计算商多项式 $h(x)=\left(s\cdot W(x)-s\cdot U(x)*s\cdot V(x)\right)/z(x)$

$$s\cdot W(x)-s\cdot U(x)*s\cdot V(x)=0$$

$$\Updownarrow$$

$$z(x)\,|\,s\cdot W(x)-s\cdot U(x)*s\cdot V(x)$$

关键结论：向量与多项式系数的组合运算等价于目标多项式整除QAP多项式

NP问题：已知目标多项式z(x)和三组多项式$w_1(x),\dots,w_n(x)$, $u_1(x),\dots,u_n(x)$, $v_1(x),\dots, v_n(x)$，求向量s， 使得目标多项式z(x)整除QAP多项式s•W(x)-s•U(x)*s•V(x)

zk-SNARK协议

ω 满足任意多项式时间运算关系 Y=F(ω) ⟺	ω 满足 R1CS约束 (电路约束) ⟺	向量s (内积) 多维向量 (u_i,v_i,w_i) ⟺	向量s (内积) 矩阵 U,V,W ⟺	向量s (组合运算) 多项式 U(x),V(x),W(x) ⟺	目标多项式 (整除) QAP多项式 (**NP问题**) ⟺	QAP多项式 目标多项式 商多项式 (多指数运算) (**证据**) ⟺	椭圆曲线 离散对数 (双线性映射) 双线性群

证明方：基于3个多项式构造椭圆曲线离散对数点

$s\cdot W(x)-s\cdot U(x)*s\cdot V(x)$	**取出系数**	$a_1,\ldots,a_m$	**构造椭圆曲线离散对数点**	$a_1\cdot G,\ldots,a_m\cdot G$
$z(x)$	**取出系数**	$z_1,\ldots,z_l$	**构造椭圆曲线离散对数点**	$z_1\cdot G,\ldots,z_l\cdot G$
$h(x)$	**取出系数**	$h_1,\ldots,h_{m-l}$	**构造椭圆曲线离散对数点**	$h_1\cdot G,\ldots,h_{m-l}\cdot G$

验证方：基于椭圆曲线离散对数点，使用双线性映射，重构整除关系，完成向量s的正确性验证

$$\vec{e}(a_1\cdot G,*)=\vec{e}(z_1\cdot G,*)\cdot\vec{e}(h_1\cdot G,*)$$

NP问题的可验证性表明：证明方天然知道解向量s，而验证方在椭圆曲线离散对数点上重构整除关系，验证NP问题的正确性，却不知道解向量s。

zk-SNARK典型应用

zk-Rollup

数据

① 需要提交一层的数据是公开数据 $statement=(1, MerkleRoot, PubData)$
② 剩余的二层所有秘密数据 $witness=(s_1,\ldots,s_n)$
③ 基于公开数据和隐私数据，构造向量 $s=(statement, witness)=(1, MerkleRoot, PubData; s_1,\ldots,s_n)$

ω

运算关系

Merkle树
哈希函数
EdDSA验证

⟺

ω
满足
R1CS约束
约束合并

⟺

向量s
（内积）
多维向量
(u_i,v_i,w_i)

⟺

向量s
（内积）
矩阵
U,V,W

⟺

向量s
（组合运算）
多项式
U(x),V(x),W(x)

⟺

目标多项式
（整除）
QAP多项式
（NP问题）

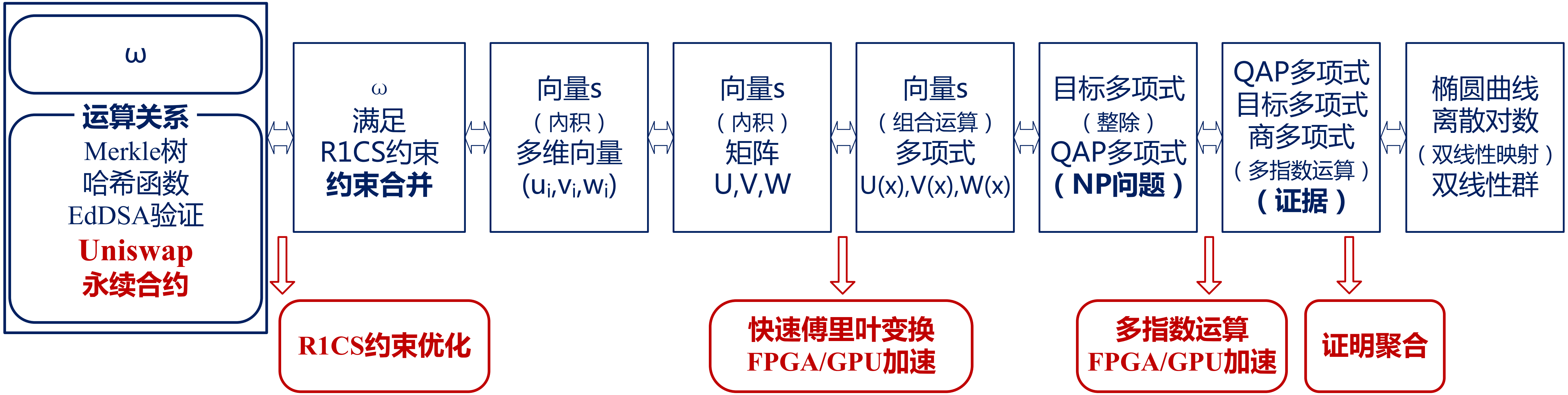
⟺

QAP多项式
目标多项式
商多项式
（多指数运算）
（证据）

⟺

椭圆曲线
离散对数
（双线性映射）
双线性群

贡献


ω
运算关系
Merkle树
哈希函数
EdDSA验证
Uniswap
永续合约
ω
满足
R1CS约束
约束合并
向量s
（内积）
多维向量
(u_i,v_i,w_i)
向量s
（内积）
矩阵
U,V,W
向量s
（组合运算）
多项式
U(x),V(x),W(x)
目标多项式
（整除）
QAP多项式
（NP问题）
QAP多项式
目标多项式
商多项式
（多指数运算）
（证据）
椭圆曲线
离散对数
（双线性映射）
双线性群
R1CS约束优化
快速傅里叶变换
FPGA/GPU加速
多指数运算
FPGA/GPU加速
证明聚合

总结与展望

固定算法

Merkle树
哈希函数
EdDSA验证
Uniswap
永续合约

⟺

电路
R1CS约束

⟺

Groth16
证明系统
CRS非通用

非固定算法

虚拟机
EVM

⟺

电路
Plonk门

⟺

Plonk
证明系统
CRS通用

谢 谢