密码学系列讲座

第 4 课：承诺、零知识证明、BulletProof 范围证明、Diffie-Hellman 密钥协商

lynndell 博士

新火科技 密码学专家 lynndell2010@gmail.com

目录

1.密码学承诺

1.1 承诺

承诺分为 3 个步骤：承诺、打开承诺、验证承诺。

承诺：发送方将某个值 x 封装为 y 发送给接收方。（1）发送方不能修改信封中的值（绑定性）；（2）接收方无法知道 x（隐藏性）。
打开承诺：发送方揭露 x。
校验承诺：接收方校验打开的值 x 与 y 中封装的 x 是否相同。

承诺一个值

- **承诺：**选择 x，计算 $y = f(x)$，发送函数值 y；
- **打开承诺：**发送原象 x；
- **校验承诺：**函数一致性 $y == f(x)$

对函数有一定要求：
- ✧ 函数求逆是 **NP 困难**的，需要**指数时间**暴力搜索。防止根据承诺值 y 计算 x。
- ✧ 但是校验简单，仅需要多项式时间计算复杂度。
- ✧ 该函数通常是哈希函数或 pedersen 承诺函数等。

承诺一个多项式

- **承诺：**选择 $n+1$ 个随机数 $a_0,...,a_n$，构造多项式 $f(x) = \sum_{i=0}^{n} a_i x^i$，计算

$$A_i = a_i \cdot G, i = 0,...,n$$

  发送 $A_i, i = 0,...,n$。

- **打开承诺：**打开一个随机点 k，计算 $f(k) = \sum_{i=0}^{n} a_i k^i$；**发送** $(k, f(k))$。

- **校验承诺：**基于 $A_i, i = 0,...,n$ 校验 $(k, f(k))$ 正确性：$f(k) \cdot G == \sum_{i=0}^{n} (k^i \cdot A_i)$。

公式推导：$f(k) \cdot G = \sum_{i=0}^{n} (a_i k^i \cdot G) = \sum_{i=0}^{n} (k^i \cdot A_i)$

如果攻击者不知道多项式，选择随机数作为函数值，则发生碰撞的概率可忽略。
因此，不必打开多项式所有系数，仅打开一个或多个**函数点**即可，从而减少发送数据。
此外，没泄露多项式，具有保密性。需要 n+1 个值，才会泄露多项式的系数。
KZG 承诺、Dan Boneh 承诺等多项式承诺在后续 zk 系列的 Plonk 证明系统中介绍。

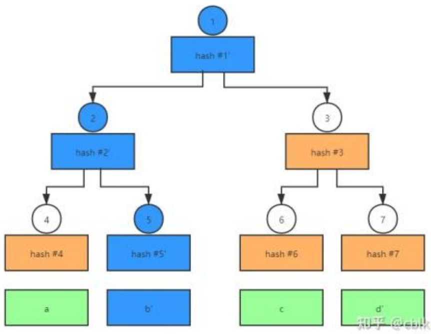
1.2 哈希承诺

- **承诺：**发送哈希值 y
- **打开承诺：**发送原象 x
- **校验承诺：**校验哈希一致性 $y == hash(x)$

哈希函数求逆满足 NP 困难。

1.3 Merkle 承诺与 Merkle 证明

- **承诺：**发送 $root$ 。
- **打开承诺：**发送叶子节点 x_i 和 $path_i$ 。其中 $path_i$ 是指**兄弟节点**。
- **校验承诺：**校验 $root == Merkle(x_i, path_i)$ 。



问题：证明方证明知道每个叶子的值 $x_i, i = 0,...,2^n$ ，树高度为 100。

低效做法：

- **承诺：**发送 $root$ ；
- **打开承诺：**发送**所有**叶子节点 $x_i, i = 0,...,2^n$ ；
- **校验承诺：**校验 $root == Merkle(x_0,...,x_{2^n})$ 。

高效做法：检测 n（例如 n=20）个点，没必要全部打开。

- **承诺：**发送 $root$ ；
- **打开承诺：**发送叶子节点 x_i 和 $path_i$ ， $i = 1,...,20$ 。
- **校验承诺：**校验 $root == Merkle(x_i, path_i), i = 1,...,20$ 。

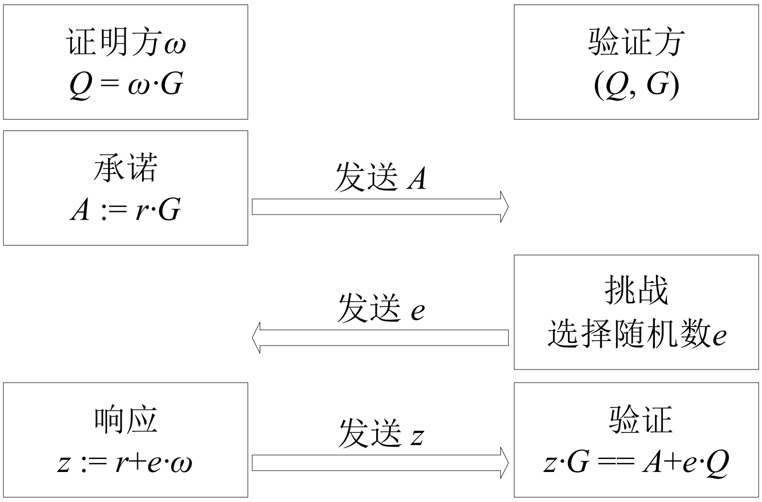
发送数据和校验复杂度均降低。

如果每个叶子的取值是 0 或 1，则 n 次均成功概率为 $1/2^{20}$ 。

如果每个叶子的取值空间为 m ，则 n 次均成功概率为 $1/m^{20}$ 。

核心思想：从概率角度，不必打开全部叶子节点；仅需要打开 n 个点，如果每次都正确，则伪造成功概率指数降低。因此，验证方相信证明方知道所有叶子节点。

1.4 Sigma 零知识证明中的承诺



承诺 $A=r\cdot G$、挑战e、响应$z=r+e\cdot\omega$、校验$z\cdot G == A+e\cdot Q$

承诺随机数 r，但是没打开 r，而是在响应中**使用 r** 随机化秘密ω。

1.5.Pedersen 承诺

初始化：椭圆曲线生成元为G,H， $H=\alpha\cdot G$，其中α保密。

- **承诺：**Token 数量为m和随机数为r，则计算$P:=m\cdot G+r\cdot H$，发送P；
- **打开承诺：**发送m和r；
- **校验承诺：**校验一致性$P == m\cdot G+r\cdot H$。

Pedersen **承诺的同态性：**
(承诺)
初始状态：Alice 和 Bob 的余额密文是 0。

- Carol 对m_1个 Token **承诺：** $P_1=m_1\cdot G+r_1\cdot H$， 接收地址为$Addr_{Alice}$，然后签名广播；

  打开承诺：私底下保密发送m_1,r_1给 Alice；Alice 校验 Pedersen 承诺的一致性，且等交易单上链后，则收款成功。

- Dave 对m_2个 Token **承诺：** $P_2=m_2\cdot G+r_2\cdot H$， 接收地址为$Addr_{Alice}$，然后签名广播；

  打开承诺：私底下保密发送m_2,r_2给 Alice；Alice 校验 Pedersen 承诺的一致性，且等交易单上链后，则收款成功。

经过**共识算法**，矿工上链 Alice 余额密文： $P_1+P_2=(m_1+m_2)\cdot G+(r_1+r_2)\cdot H$

Alice 知道秘密：m1+m2 和随机数为 r1+r2，则 Alice 能花费该费用。

- Alice 对 m_3 个 Token **承诺：** $P_3 = m_3 \cdot G + r_3 \cdot H$，接收地址为 $Addr_{Bob}$，然后签名广播；

  打开承诺： 私底下保密发送 m_3, r_3 给 Bob；Bob 校验 Pedersen 承诺的一致性，且等交易单上链后，则收款成功。

经过**共识算法**，矿工上链 Alice 和 Bob 的余额密文：

$$Alice: P_1 + P_2 - P_3 = (m_1 + m_2 - m_3) \cdot G + (r_1 + r_2 - r_3) \cdot H$$
$$Bob: P_3 = m_3 \cdot G + r_3 \cdot H$$

- Alice 知道 $m_1 + m_2 - m_3, r_1 + r_2 - r_3$ 可以继续支付；
- Bob 知道 m_3, r_3 也可以继续支付。

如果 α 泄露：

$H = a \cdot G$

Alice 知道 G, H 之间的离散对数 α，后果很严重。

真实情况： Alice 拥有小金额 m=10 和随机数 r，**余额承诺为** $P = m \cdot G + r \cdot H$。

Alice 能够计算 α^{-1}，选择一个大金额 m′=200000，计算**随机数** $r' = r - (m' - m)\alpha^{-1}$。

- Alice 支付 m′个 Token，**支付承诺为** $P == m' \cdot G + r' \cdot H$，接收地址为 $Addr_{Bob}$，然后签名广播；**打开承诺：** 私底下保密发送 m', r' 给 Bob；Bob 校验 Pedersen 承诺的一致性，且等交易单上链后，则收款成功。

经过**共识算法**，矿工上链 Bob 余额承诺：$P == m' \cdot G + r' \cdot H$

公式推导：

$$
\begin{aligned}
& m' \cdot G + r' \cdot H = m' \cdot G + \left(r - (m' - m)\alpha^{-1}\right) \cdot H \\
& = m' \cdot G + r \cdot H - m'\alpha^{-1} \cdot H + m\alpha^{-1} \cdot H \\
& = m \cdot G + r \cdot H \\
& = P
\end{aligned}
$$

类似结论：

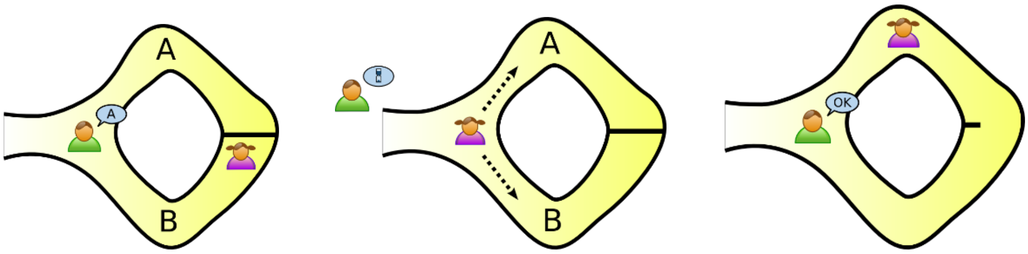
zcash 中各个生成元 $g_1, \ldots, g_n, h_1, \ldots, h_n$ 之间的**离散对数**不能泄露；

Plonk 中的 KZG 承诺 CRS 使用的随机数 α 不能泄露；

如果**泄露**，则后果很严重。

2.零知识证明

概率游戏：



- 游戏执行 1 次，小红正确的概率为$1/2$，可能碰巧成功。
- 游戏执行 n 次，小红全正确的概率为$1/2^n$，呈指数降低。
- 因此，如果小红每次都正确，**不会是碰巧，而是知道开门秘诀！**

✧ 阿里巴巴 Cave 要求：没有其他通道（没后门）。
✧ zkSnark 初始化设置：要求 n 参与方参与初始化，没后门。

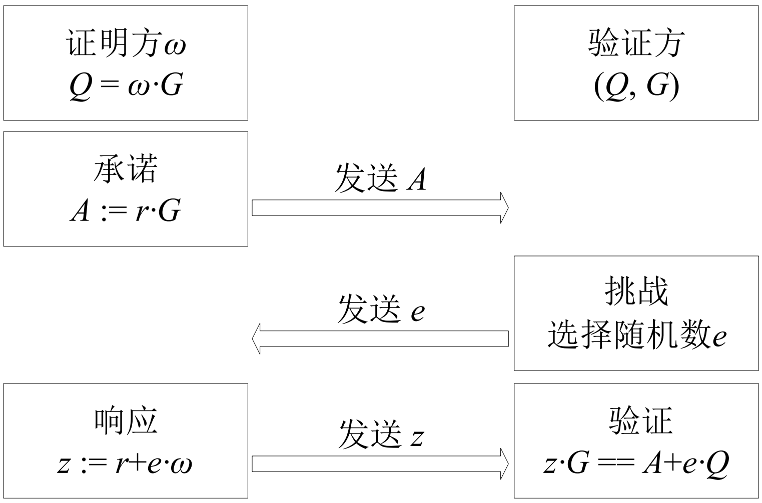
2.1 Sigma 零知识证明协议

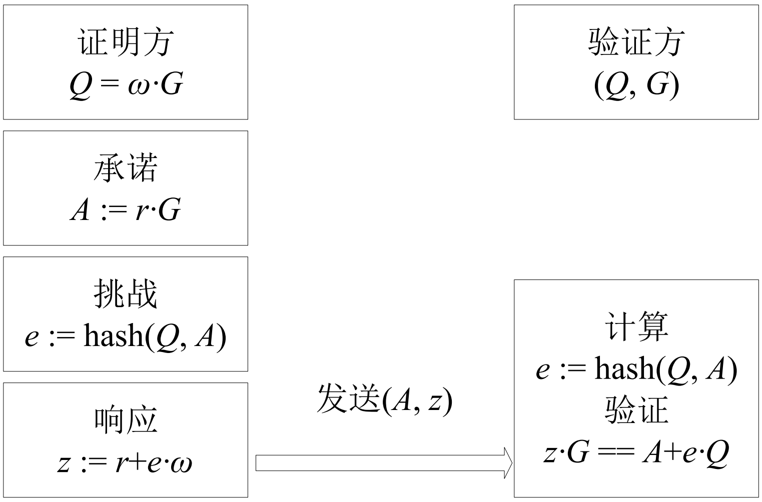
Sigma 零知识证明：知道秘密 ω，且与公开输入 Q 满足离散对数关系 $Q=\omega\cdot G$。

1：**（承诺）** P 选择随机数 r，计算 $A=r\cdot G$，发送 A；
2：**（挑战）** V 发送随机数 e；
3：**（响应）** P 计算 $z=r+e\cdot\omega$；发送 z；

4：**（验证）** V 校验 $z\cdot G==A+e\cdot Q$。

公式推导：$z\cdot G=(r+e\omega)\cdot G=A+e\cdot Q$

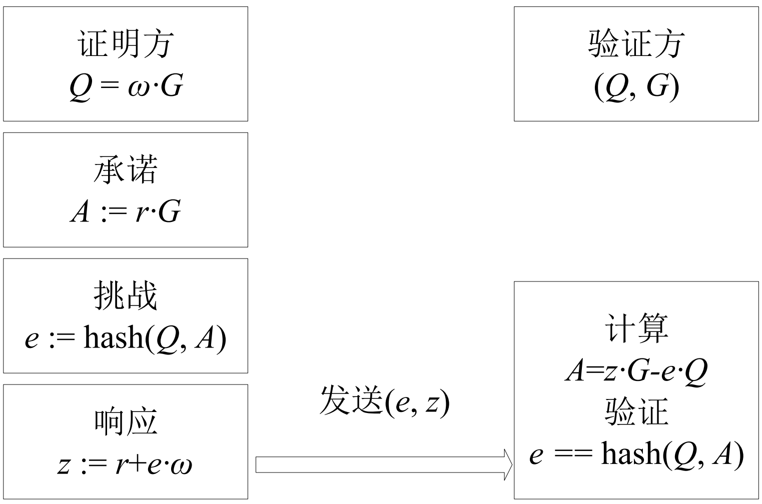




非交互式 A 版：Sigma 零知识证明，发送**承诺**和**响应（数据量较大）。**
验证方重新计算挑战，然后校验等式。
额外条件：非交互式 A 版零知识证明要求哈希函数的输出是抗碰撞的!

哈希函数 $hash(X) = Y$，要求不能在多项式时间内找到 X'，满足 $hash(X') = Y$。

$$hash(Q, A) = e$$
$$hash(Q, A') = e$$
$$A = z \cdot G - e \cdot Q$$
$$A' = z' \cdot G - e \cdot Q$$



非交互式 B 版：Sigma 零知识证明，发送**挑战**和**响应（数据量较小）。**
验证方重新计算承诺，然后哈希校验。
额外条件：非交互式 B 版零知识证明要求哈希函数的输出是抗碰撞的!

2.2 Sigma 协议应用到 ElGamal 同态加密

2.2.1 密文同态运算

1. **密钥生成：**Alice/Bob/Carol/Dave 私钥和公钥分别为

$$
\begin{aligned}
&Alice:(\alpha_1,g_1),g_1=g^{\alpha_1}\\
&Bob:(\alpha_2,g_2),g_2=g^{\alpha_2}\\
&Carol:(\alpha_3,g_3),g_3=g^{\alpha_1}\\
&Dave:(\alpha_4,g_4),g_4=g^{\alpha_2}
\end{aligned}
$$

余额初始状态：

有多少 Token

- Alice 余额初始状态密文为 $[C_0,D_0]$，其中 $C_0=g^{r_0},D_0=g^{m_0}\cdot g_1^{r_0}$
- Bob 余额初始状态密文为 $[C_0',D_0']$，其中 $C_0'=g^{r_0'},D_0'=g^{m_0'}\cdot g_2^{r_0'}$

余额，明文

- Carol/Dave 起始状态金额为 0。

2. Alice 支付给 Carol 金额数量为 m_1，使用 ElGamal 同态加密生成密文 $[C_1,D_1],[C_1',D_1']$，

并生成零知识证明 $zkProof$ 和范围证明 $BulletProof$，

- Alice 选择随机数 r_1，基于 Carol 的公钥 g_3，生成 $[C_1,D_1]$；
- Alice 选择随机数 r_1'，基于自己的公钥 g_1，生成 $[C_1',D_1']$；

$$
\begin{aligned}
&C_1=g^{r_1},D_1=g^{m_1}\cdot g_3^{r_1}\\
&C_1'=g^{r_1'},D_1'=g^{m_1'}\cdot g_1^{r_1'}\\
&ZK\{r_1,r_1',m_1,m_1',m_1=m_1'|C_1,D_1,C_1',D_1'\}\\
&BulletProof\{0\le m_0-m_1\le 2^{32},0\le m_1\le 2^{32}\}
\end{aligned}
$$

Alice 对交易单签名；**矿工**验证签名、零知识证明和范围证明。然后：

- **更新** Carol 金额状态记为 $[C_1,D_1]$，则 Carol 能够解密获得 m_1 并进行支付。解密过程：$g^{m_1}:=D_1/(C_1)^{\alpha_3}$；　m_1 : 32-bit，能暴力搜索出来
- **更新** Alice 的金额状态记为 $[C_0/C_1',D_0/D_1']$，则 Alice 能够解密获得 m_0-m_1 并进行支付。解密过程：$g^{m_0-m_1'}:=(D_0/D_1')/(C_0/C_1')^{\alpha_1}$。

Carol 余额增加 m_1，Alice 余额减小 m_1'，所有需要 $ZK\{r_1,r_1',m_1,m_1',m_1=m_1'|C_1,D_1,C_1',D_1'\}$。

3. Bob 支付给 Carol 金额数量为 m_2，使用 ElGamal 同态加密生成密文 $[C_2,D_2],[C_2',D_2']$，

并生成零知识证明和范围证明

$$C_2 = g^{r_2}, D_2 = g^{m_2} \cdot g_3^{r_2}$$
$$C_2' = g^{r_2'}, D_2' = g^{m_2'} \cdot g_2^{r_2'}$$
$$ZK\{r_2, r_2', m_2, m_2', m_2 = m_2' | C_2, D_2, C_2', D_2'\}$$
$$BulletProof\{0 \le m_0' - m_2 \le 2^{32}, 0 \le m_2 \le 2^{32}\}$$

Bob 对交易单签名；**矿工**验证签名、验证零知识证明和范围证明，然后：

- **更新** Carol 金额状态记为$[C_1 \cdot C_2, D_1 \cdot D_2]$，则 Carol 能够解密获得$m_1 + m_2$并进行支付。解密过程：$g^{m_1+m_2} := D_1 \cdot D_2 / (C_1 \cdot C_2)^{\alpha_3}$
- **更新** Bob 的金额状态记为$[C_0'/C_2', D_0'/D_2']$，则 Bob 能够解密获得$m_0' - m_2$并进行支付。解密过程：$g^{m_0'-m_2'} := (D_0'/D_2')/(C_0'/C_2')^{\alpha_2}$

4. Carol 支付给 Dave 金额数量为m_3，使用 ElGamal 同态加密生成密文$[C_3, D_3],[C_3', D_3']$，并生成零知识证明和范围证明

$$C_3 = g^{r_3}, D_3 = g^{m_3} \cdot g_4^{r_3}$$
$$C_3' = g^{r_3'}, D_3' = g^{m_3'} \cdot g_3^{r_3'}$$
$$ZK\{r_3, r_3', m_3, m_3', m_3 = m_3' | C_3, D_3, C_3', D_3'\}$$
$$BulletProof\{0 \le m_1 + m_2 - m_3 \le 2^{32}, 0 \le m_3 \le 2^{32}\}$$

Carol 对交易单签名；**矿工**验证签名、验证零知识证明和范围证明，然后：

- **更新** Carol 金额状态记为$[C_1 \cdot C_2 / C_3', D_1 \cdot D_2 / D_3']$，则 Carol 能够解密获得$m_1 + m_2 - m_3$并进行支付。

  解密过程：$g^{m_1+m_2-m_3} := (D_1 \cdot D_2 / D_3')/(C_1 \cdot C_2 / C_3')^{\alpha_3}$。
- **更新** Dave 的金额状态记为$[C_3, D_3]$，则 Dave 能够解密获得m_3并进行支付。

  解密过程：$g^{m_3} := (D_3)/(C_3)^{\alpha_4}$。

Sigma 零知识证明确保：支付方的减少额 == 接收方的增加额。

2.2.2 Sigma 证明 2 个值相等

Alice：

选择随机数 r1，使用 Carol 的公钥 g3 和金额 m1 生成 ElGamal 加密密文 C1,D1；
选择随机数 r2，使用自己的公钥 g1 和金额 m2 生成 ElGamal 加密密文 C2,D2；

$$C_1 = g^{r_1}, D_1 = g^{m_1} \cdot g_3^{r_1}$$
$$C_1' = g^{r_1'}, D_1' = g^{m_1'} \cdot g_1^{r_1'}$$
$$ZK\{r_1, r_1', m_1, m_1', m_1 = m_1' | C_1, D_1, C_1', D_1'\}$$

符号修改：令 $m_2 = m_1'$。

Alice 需要证明其知道 $r_1, r_2, m_1, m_2, m_1 = m_2$，满足以下 4 个离散对数关系：

$$C_1 = g^{r_1}, D_1 = g^{m_1} \cdot g_3^{r_1}，$$ 接收方的增加额
$$C_2 = g^{r_2}, D_2 = g^{m_2} \cdot g_1^{r_2}$$ 支付方的减少额

情况 1： 假如 $m_1 \neq m_2$， Alice 选择 m_1 作为 Sigma 零知识证明的输入。

承诺： 选择 2 个随机数 $r_1', r_2' \in \mathbb{Z}_n$ 和金额 m_1， 构造 2 个对应的密文

$$C_1' = g^{r_1'}, D_1' = g^{m_1} \cdot g_3^{r_1'}$$
$$C_2' = g^{r_2'}, D_2' = g^{m_1} \cdot g_1^{r_2'}$$

挑战： 计算哈希值 $e := hash(g_1, g_3, C_1, D_1, C_2, D_2, C_1', D_1', C_2', D_2')$

响应： 计算 $z_1 := r_1' + e \cdot r_1, z_2 := r_2' + e \cdot r_2, \tilde{m} := m_1 + e \cdot m_1$

发送数据为： $Proof\{\mathrm{C_1', D_1', C_2', D_2', z_1, z_2, \tilde{m}}\}$

验证： 计算哈希值 $e := hash(g_1, g_3, C_1, D_1, C_1', D_1', C_1', D_1', C_2', D_2')$， 校验以下 4 个等式的一致性

$$(C_1)^e \cdot C_1' = g^{z_1}，\ (D_1)^e \cdot D_1' = g^{\tilde{m}} \cdot g_3^{z_1}$$

$$(C_2)^e \cdot C_2 = g^{z_2}，\ (D_2)^e \cdot D_2' = g^{\tilde{m}} \cdot g_1^{z_2}$$

公式推导：

$$(C_1)^e \cdot C_1' = g^{e \cdot r_1} \cdot g^{r_1'} = g^{z_1}，\ (D_1)^e \cdot D_1' = \left(g^{e \cdot m_1} \cdot g_3^{e \cdot r_1}\right) \cdot \left(g^{m_1} \cdot g_3^{r_1'}\right) = g^{\tilde{m}} \cdot g_3^{z_1}$$

$$(C_2)^e \cdot C_2' = g^{e \cdot r_2} \cdot g^{r_2'} = g^{z_2}，\ (D_2)^e \cdot D_2' = \left(g^{e \cdot m_2} \cdot g_1^{e \cdot r_2}\right) \cdot \left(g^{m_1} \cdot g_1^{r_2'}\right) \neq g^{\tilde{m}} \cdot g_1^{z_2}$$ 失败

情况 2： 假如 $m_1 \neq m_2$， 证明方选择 m_2 作为 Sigma 零知识证明的输入。

承诺： 选择 2 个随机数 $r_1', r_2' \in \mathbb{Z}_n$ 和金额 m_2， 构造 2 个对应的密文

$$C_1' = g^{r_1'}, D_1' = g^{m_2} \cdot g_3^{r_1'}$$
$$C_2' = g^{r_2'}, D_2' = g^{m_2} \cdot g_1^{r_2'}$$

挑战： 计算哈希值 $e := hash(g_1, g_3, C_1, D_1, C_2, D_2, C_1', D_1', C_2', D_2')$

响应：计算 $z_1 := r_1' + e \cdot r_1, z_2 := r_2' + e \cdot r_2, \tilde{m} := m_2 + e \cdot m_2$

发送数据为： $Proof\{C_1', D_1', C_2', D_2', z_1, z_2, \tilde{m}\}$

验证：计算哈希值 $e := hash(g_1, g_3, C_1, D_1, C_1', D_1', C_1', D_1', C_2', D_2')$，校验以下 4 个等式的一致性

$$(C_1)^e \cdot C_1' = g^{z_1}, \quad (D_1)^e \cdot D_1' \neq g^{\tilde{m}} \cdot g_3^{z_1} \text{ 失败}$$

$$(C_2)^e \cdot C_2 = g^{z_2}, \quad (D_2)^e \cdot D_2' = g^{\tilde{m}} \cdot g_1^{z_2}$$

情况 3：假如 $m_1 \neq m_2$，证明方选择 m_1, m_2 作为 Sigma 零知识证明的输入。

承诺：选择 2 个随机数 $r_1', r_2' \in \mathbb{Z}_n$ 和金额 m_2，构造 2 个对应的密文

$$C_1' = g^{r_1'}, D_1' = g^{m_1} \cdot g_3^{r_1'}$$
$$C_2' = g^{r_2'}, D_2' = g^{m_2} \cdot g_1^{r_2'}$$

挑战：计算哈希值 $e := hash(g_1, g_3, C_1, D_1, C_2, D_2, C_1', D_1', C_2', D_2')$

响应：计算 $z_1 := r_1' + e \cdot r_1, z_2 := r_2' + e \cdot r_2, \tilde{m} := m_1 + e \cdot m_2$

发送数据为： $Proof\{C_1', D_1', C_2', D_2', z_1, z_2, \tilde{m}\}$

验证：计算哈希值 $e := hash(g_1, g_3, C_1, D_1, C_1', D_1', C_1', D_1', C_2', D_2')$，校验以下 4 个等式的一致性

$$(C_1)^e \cdot C_1' = g^{z_1}, \quad (D_1)^e \cdot D_1' \neq g^{\tilde{m}} \cdot g_3^{z_1} \text{ 失败}$$

$$(C_2)^e \cdot C_2 = g^{z_2}, \quad (D_2)^e \cdot D_2' \neq g^{\tilde{m}} \cdot g_1^{z_2} \text{ 失败}$$

情况 4：假如 2 个同态密文中 $m_1 = m_2$，证明方选择 $m = m_1 = m_2$ 作为 Sigma 零知识证明的输入。

承诺：选择 2 个随机数 $r_1', r_2' \in \mathbb{Z}_n$ 和金额 m，构造 2 个对应的密文

$$C_1' = g^{r_1'}, D_1' = g^{m} \cdot g_3^{r_1'}$$
$$C_2' = g^{r_2'}, D_2' = g^{m} \cdot g_1^{r_2'}$$

挑战：计算哈希值 $e := hash(g_1, g_3, C_1, D_1, C_2, D_2, C_1', D_1', C_2', D_2')$

公开　挑战，能算出来

响应：计算 $z_1 := r_1' + e \cdot r_1, z_2 := r_2' + e \cdot r_2, \tilde{m} := m + e \cdot m$

发送数据为： $Proof\{C_1', D_1', C_2', D_2', z_1, z_2, \tilde{m}\}$ **协议缺点：** $\tilde{m}$ 会泄露 m_1

验证： 计算哈希值 $e := hash(g_1, g_3, C_1, D_1, C_1', D_1', C_1', D_1', C_2', D_2')$，校验以下 4 个等式的一致性

$$(C_1)^e \cdot C_1' = g^{z_1},\ \ (D_1)^e \cdot D_1' = g^{\tilde{m}} \cdot g_3^{z_1}$$

$$(C_2)^e \cdot C_2 = g^{z_2},\ \ (D_2)^e \cdot D_2' = g^{\tilde{m}} \cdot g_1^{z_2}$$

情况 5： 假如 $m_1 = m_2$，证明方选择随机数 m_0 作为 Sigma 零知识证明的输入。

承诺： 选择 2 个随机数 $r_1', r_2' \in \mathbb{Z}_n$ 和随机数 m_0，构造 2 个对应的密文

$$C_1' = g^{r_1'}, D_1' = g^{m_0} \cdot g_3^{r_1'}$$
$$C_2' = g^{r_2'}, D_2' = g^{m_0} \cdot g_1^{r_2'}$$

挑战： 计算哈希值 $e := hash(g_1, g_3, C_1, D_1, C_2, D_2, C_1', D_1', C_2', D_2')$

响应： 计算 $z_1 := r_1' + e \cdot r_1, z_2 := r_2' + e \cdot r_2, \tilde{m} := m_0 + e \cdot m_1$

发送数据为： $Proof\{\mathrm{C_1', D_1', C_2', D_2', z_1, z_2, \tilde{m}}\}$

验证： 计算哈希值 $e := hash(g_1, g_3, C_1, D_1, C_1', D_1', C_1', D_1', C_2', D_2')$，校验以下 4 个等式的一致性

$$(C_1)^e \cdot C_1' = g^{z_1},\ \ (D_1)^e \cdot D_1' = g^{\tilde{m}} \cdot g_3^{z_1}$$

$$(C_2)^e \cdot C_2 = g^{z_2},\ \ (D_2)^e \cdot D_2' = g^{\tilde{m}} \cdot g_1^{z_2}$$

2 条验证等式中的 $\tilde{m}$ 对应 m_1 和 m_2，响应等式中的 $\tilde{m}$ 也对应 m_1 或 m_2，因此，$m_1 = m_2$。

因此，Sigma 零知识证明确保：支付方的减少额 == 接收方的增加额。
还有作恶方法：
余额 0 Token，支付 100 Token。接收方增加 100 Token，发送方余额为-100 Token。
因此，需要 **BulletProof 范围证明**确保支付 Token 数量与余额 Token 数量大于或等于零。

3.BulletProof 范围证明

取值范围

3.1 符号说明

循环群 $\mathbb{G}$ 的阶为 p，环 $\mathbb{Z}_p$ 的阶为 p。$\mathbb{G}^n$ 和 $\mathbb{Z}_p^n$ 是对应的 n 为向量。$\mathbb{Z}_p^*$ 是指 $\mathbb{Z}_p / \{0\}$。

$\mathbb{G}$ 的 4 个生成元为 g, h, v, u。n 维向量 $\vec{a} = \{a_1, ..., a_n\}$。

n 行 m 列**矩阵** $\vec{A}\in\mathbb{F}^{n\times m}$ 的第 i 行 j 列的元素为 $a_{i,j}$。

标量 $c\in\mathbb{Z}_p$ 与向量的 $\vec{a}\in\mathbb{Z}_p^n$ 的**乘积**为 $\vec{b}=c\cdot\vec{a}=\{c\cdot a_1,...,c\cdot a_n\}\in\mathbb{Z}_p^n$。

两个**向量内积** $\langle\vec{a},\vec{b}\rangle=\sum_{i=1}^n a_ib_i$ 是一个值。

向量 **Hadamard 乘积** $\vec{a}\circ\vec{b}=(a_1b_1,...,a_nb_n)\in\mathbb{F}^n$ 是一个向量。

向量多项式 $p(X)=\sum_{i=0}^d \vec{p}_iX^i\in\mathbb{Z}_p^n$，系数向量为 $\vec{p}_i=(p_1,...,p_n)\in\mathbb{Z}_p^n$。

向量多项式的内积 $t(X)=\langle l(X),r(X)\rangle=\sum_{i=0}^d\sum_{j=0}^i\langle\vec{l}_i,\vec{r}_j\rangle X^{i+j}\in\mathbb{Z}_p[X]$ 是一个多项式。

生成元向量 $\vec{g}=(g_1,...,g_n)\in\mathbb{G}^n$，随机数向量 $\vec{a}=\{a_1,...,a_n\}\in\mathbb{Z}_p^n$，则承诺

$$C=\vec{g}^{\vec{a}}=\prod_{i=1}^n g_i^{a_i}\in\mathbb{G}$$

该承诺具有**绑定性**，没有隐藏性（因为缺少随机项）。

对于**承诺** $C=\vec{g}^{\vec{a}}=\prod_{i=1}^n g_i^{a_i}\in\mathbb{G}$，令 $g_i'=g_i^{b_i^{-1}}$，则 $C=\prod_{i=1}^n (g_i')^{a_ib_i}\in\mathbb{G}$。

$if\,(\vec{a}\in\mathbb{Z}_p^n,\vec{b}\in\mathbb{Z}_p^m),then(\vec{a}\,\|\,\vec{b}\in\mathbb{Z}_p^{n+m})$。

连接

$\vec{a}_{[:l]}=(a_1,...,a_l)\in\mathbb{F}^l,\vec{a}_{[l:]}=(a_{l+1},...,a_n)\in\mathbb{F}^{n-l}$，$n=2^k$。n/2=$l$

$k\in\mathbb{Z}_p^*$，$\vec{k}^n=(1,k,k^2,k^3,...,k^{n-1})\in(\mathbb{Z}_p^*)^n,\vec{k}^{-n}=(1,k^{-1},k^{-2},k^{-3},...,k^{-(n-1)})\in(\mathbb{Z}_p^*)^n$。

例如 $2\in\mathbb{Z}_p^*,\vec{2}^n=(1,2,2^2,2^3,...,2^{n-1})\in(\mathbb{Z}_p^*)^n,\vec{2}^{-n}=(1,2^{-1},2^{-2},2^{-3},...,2^{-(n-1)})\in(\mathbb{Z}_p^*)^n$。

对于生成元向量： $\vec{g}=(g_1,...,g_n)\in\mathbb{G}^n,\vec{h}=(h_1,...,h_n)\in\mathbb{G}^n$，其中 $n=2^k$。

核心结论：以下 k+3 个运算关系 ①,②,③,$\boxed{1},\boxed{2},\boxed{3},...,\boxed{k-1},\boxed{k}$ **等价。**

对于 P,c 是公开参数，证明方证明知道**秘密向量** $\vec{a},\vec{b}\in\mathbb{Z}_p^n$，满足

$$P_1=\vec{g}^{\vec{a}}\vec{h}^{\vec{b}},c=\langle\vec{a},\vec{b}\rangle \text{ 运算关系①}$$

函数值

左边等式是 Pedersen 承诺的一般化。

如果完全打开向量 $\vec{a},\vec{b}\in\mathbb{Z}_p^n$，验证方能够校验。但是，发送数据量为 2n。

构造一个与上述等价的新关系

$$P_1=\vec{g}^{\vec{a}}\vec{h}^{\vec{b}}\cdot u^{\langle\vec{a},\vec{b}\rangle} \text{ 运算关系②}$$

也是一个生成元

令 $n'=n/2$， $\vec{a}_1,\vec{a}_2,\vec{b}_1,\vec{b}_2\in\mathbb{Z}_p^{n'}$， 定义同态哈希函数 $Hash$

$$Hash(\vec{a}_1,\vec{a}_2,\vec{b}_1,\vec{b}_2,c)=\vec{g}_{[:n']}^{\vec{a}_1}\cdot\vec{g}_{[n':]}^{\vec{a}_2'}\cdot\vec{h}_{[:n']}^{\vec{b}_1}\cdot\vec{h}_{[n':]}^{\vec{b}_2'}\cdot u^c$$

这是 pedersen 承诺的一般化。

因此，有以下同态性：

底数相乘，指数相加

$$Hash(\vec{a}_1,\vec{a}_1',\vec{b}_1,\vec{b}_1',c_1)\cdot Hash(\vec{a}_2,\vec{a}_2',\vec{b}_2,\vec{b}_2',c_2)=Hash(\vec{a}_1+\vec{a}_2,\vec{a}_1'+\vec{a}_2',\vec{b}_1+\vec{b}_2,\vec{b}_1'+\vec{b}_2',c_1+c_2)$$

因此，运算关系②表达为以下

$$P_1=Hash\left(\vec{a}_{[:n']},\vec{a}_{[n':]},\vec{b}_{[:n']},\vec{b}_{[n':]},\left\langle\vec{a},\vec{b}\right\rangle\right)\text{运算关系③}$$

公式推导：

$$\begin{aligned}P_1&=Hash\left(\vec{a}_{[:n']},\vec{a}_{[n':]},\vec{b}_{[:n']},\vec{b}_{[n':]},\left\langle\vec{a},\vec{b}\right\rangle\right)\\&=\vec{g}_{[:n']}^{\vec{a}_{[:n']}}\cdot\vec{g}_{[n':]}^{\vec{a}_{[n':]}}\cdot\vec{h}_{[:n']}^{\vec{b}_{[:n']}}\cdot\vec{h}_{[n':]}^{\vec{b}_{[n':]}}\cdot u^{\left\langle\vec{a},\vec{b}\right\rangle}\\&=\vec{g}^{\vec{a}}\cdot\vec{h}^{\vec{b}}\cdot u^{\left\langle\vec{a},\vec{b}\right\rangle}\end{aligned}$$

3.2 向量内积承诺

功能：响应不断折半。zcash Halo2 证明系统也是使用该方法。

3.2.1 向量内积承诺 / 基于向量内积的 Sigma 零知识证明

第 1 轮： 证明方知道 n 维的秘密向量 $\vec{a},\vec{b}$，满足

$$P_1=Hash\left(\vec{a}_{[:n']},\vec{a}_{[n':]},\vec{b}_{[:n']},\vec{b}_{[n':]},\left\langle\vec{a},\vec{b}\right\rangle\right)\text{运算关系}\boxed{1}$$

$\vec{a}$的低部分为0

承诺：

$$L_1=Hash\left(0^{n'},\vec{a}_{[:n']},\vec{b}_{[n':]},0^{n'},\left\langle\vec{a}_{[:n']},\vec{b}_{[n':]}\right\rangle\right)=\vec{g}_{[:n']}^{0^{n'}}\cdot\vec{g}_{[n':]}^{\vec{a}_{[:n']}}\cdot\vec{h}_{[:n']}^{\vec{b}_{[n':]}}\cdot\vec{h}_{[n':]}^{0^{n'}}\cdot u^{\left\langle\vec{a}_{[:n']},\vec{b}_{[n':]}\right\rangle}$$

$$R_1=Hash\left(\vec{a}_{[n':]},0^{n'},0^{n'},\vec{b}_{[:n']},\left\langle\vec{a}_{[n':]},\vec{b}_{[:n']}\right\rangle\right)=\vec{g}_{[:n']}^{\vec{a}_{[n':]}}\cdot\vec{g}_{[n':]}^{0^{n'}}\cdot\vec{h}_{[:n']}^{0^{n'}}\cdot\vec{h}_{[n':]}^{\vec{b}_{[:n']}}\cdot u^{\left\langle\vec{a}_{[n':]},\vec{b}_{[:n']}\right\rangle}$$

$\vec{a}$的高部分为0

挑战： 计算随机数 $x_1=\mathrm{SHA3}(P_1,L_1,R_1)\bmod p\in\mathbb{Z}_p$。

折半响应： $\vec{a}'=x_1\vec{a}_{[:n']}+x_1^{-1}\vec{a}_{[n':]},\vec{b}'=x_1\vec{b}_{[:n']}+x_1^{-1}\vec{b}_{[n':]}$；**发送承诺 L_1,R_1 和响应 $\vec{a}',\vec{b}'$**

验证： 校验一致性 $L_1^{(x_1^2)}\cdot P_1\cdot R_1^{(x_1^{-2})}==Hash\left(x_1^{-1}\vec{a}',x_1\vec{a}',x_1\vec{b}'x_1^{-1}\vec{b}',\left\langle\vec{a}',\vec{b}'\right\rangle\right)$ 运算关系 $\boxed{2}$

公式推导：

Pedersen 哈希

$$Hash(x_1^{-1}\vec{a}',x_1\vec{a}',x_1\vec{b}',x_1^{-1}\vec{b}',\langle\vec{a}',\vec{b}'\rangle)=\vec{g}_{[:n']}^{x_1^{-1}\vec{a}'}\cdot\vec{g}_{[n':]}^{x_1\vec{a}'}\cdot\vec{h}_{[:n']}^{x_1\vec{b}'}\cdot\vec{h}_{[n':]}^{x_1^{-1}\vec{b}'}\cdot u^{\langle\vec{a}',\vec{b}'\rangle}$$

$$=\vec{g}_{[:n']}^{x_1^{-1}(x_1\vec{a}_{[:n']}+x_1^{-1}\vec{a}_{[n':]})}\cdot\vec{g}_{[n':]}^{x_1(x_1\vec{a}_{[:n']}+x_1^{-1}\vec{a}_{[n':]})}\cdot\vec{h}_{[:n']}^{x_1(x_1\vec{b}_{[:n']}+x_1^{-1}\vec{b}_{[n':]})}\cdot\vec{h}_{[n':]}^{x_1^{-1}(x_1\vec{b}_{[:n']}+x_1^{-1}\vec{b}_{[n':]})}\cdot u^{\langle\vec{a}',\vec{b}'\rangle}$$

$$L_1^{(x_1^2)}=Hash\left(0^{n'},\vec{a}_{[:n']},\vec{b}_{[n':]},0^{n'},\left\langle\vec{a}_{[:n']},\vec{b}_{[n':]}\right\rangle\right)^{(x_1^2)}=\left(\vec{g}_{[:n']}^{0^{n'}}\cdot\vec{g}_{[n':]}^{\vec{a}_{[:n']}}\cdot\vec{h}_{[:n']}^{\vec{b}_{[n':]}}\cdot\vec{h}_{[n':]}^{0^{n'}}\cdot u^{\langle\vec{a}_{[:n']},\vec{b}_{[n':]}\rangle}\right)^{(x_1^2)}$$

$$P_1=Hash\left(\vec{a}_{[:n']},\vec{a}_{[n':]},\vec{b}_{[:n']},\vec{b}_{[n':]},\left\langle\vec{a},\vec{b}\right\rangle\right)=\vec{g}_{[:n']}^{\vec{a}_{[:n']}}\cdot\vec{g}_{[n':]}^{\vec{a}_{[n':]}}\cdot\vec{h}_{[:n']}^{\vec{b}_{[:n']}}\cdot\vec{h}_{[n':]}^{\vec{b}_{[n':]}}\cdot u^{\langle\vec{a},\vec{b}\rangle}$$

$$R_1^{(x_1^{-2})}=Hash\left(\vec{a}_{[n':]},0^{n'},0^{n'},\vec{b}_{[:n']},\left\langle\vec{a}_{[n':]},\vec{b}_{[:n']}\right\rangle\right)^{(x_1^{-2})}=\left(\vec{g}_{[:n']}^{\vec{a}_{[n':]}}\cdot\vec{g}_{[n':]}^{0^{n'}}\cdot\vec{h}_{[:n']}^{0^{n'}}\cdot\vec{h}_{[n':]}^{\vec{b}_{[:n']}}\cdot u^{\langle\vec{a}_{[n':]},\vec{b}_{[:n']}\rangle}\right)^{(x_1^{-2})}$$

第 2 轮： 证明方知道 $n/2$ 维的秘密向量 $\vec{a}',\vec{b}'$，满足

$$P_2=L_1^{(x_1^2)}\cdot P_1\cdot R_1^{(x_1^{-2})}=Hash\left(x_1^{-1}\vec{a}',x_1\vec{a}',x_1\vec{b}'x_1^{-1}\vec{b}',\left\langle\vec{a}',\vec{b}'\right\rangle\right)$$ 运算关系 $\boxed{2}$

证明方发送的**承诺**和**折半响应：** 为 $L_2,R_2,\vec{a}'',\vec{b}''$。其中，$\vec{a}'',\vec{b}''\in\mathbb{Z}_p^{n/4}$

验证方需要校验 $L_2^{(x_2^2)}P_2\cdot R_2^{(x_2^{-2})}==Hash\left(x_2^{-2}\vec{a}'',x_2\vec{a}'',x_2\vec{b}'',x_2^{-2}\vec{b}'',\left\langle\vec{a}'',\vec{b}''\right\rangle\right)$ 运算关系 $\boxed{3}$

其中，x_2 为第 2 轮计算的随机数。

以此类推，响应不断折半

第 k 轮： 证明方知道折半的秘密向量 $\vec{a}^{(k-1)},\vec{b}^{(k-1)}$，满足运算关系 $\boxed{k-1}$

$$P_k=L_{k-1}^{(x_{k-1}^2)}\cdot P_{k-1}\cdot R_{k-1}^{(x_{k-1}^{-2})}=Hash\left(x_{k-1}^{-1}\vec{a}^{(k-1)},x_{k-1}\vec{a}^{(k-1)},x_{k-1}\vec{b}^{(k-1)},x_{k-1}^{-1}\vec{b}^{(k-1)},\left\langle\vec{a}^{(k-1)},\vec{b}^{(k-1)}\right\rangle\right)$$

第 k 次折半，**向量折为常量：** $\vec{a}^{(k)}=a,\vec{b}^{(k)}=b$

第 k 轮的挑战值为 x_k。

证明方需要发送的**承诺**和**折半响应：** 为 L_k,R_k,a,b。其中 $a,b\in\mathbb{Z}_p$。

验证方**校验**

$$L_k^{(x_k^2)}\cdot P_k\cdot R_k^{(x_k^{-2})}==Hash\left(x_k^{-1}\vec{a}^{(k)},x_k\vec{a}^{(k)},x_k\vec{b}^{(k)},x_k^{-1}\vec{b}^{(k)},\left\langle\vec{a}^{(k)},\vec{b}^{(k)}\right\rangle\right)$$

$$==Hash\left(x_k^{-1}a,x_ka,x_kb,x_k^{-1}b,\langle a,b\rangle\right)$$ 运算关系 $\boxed{k}$

最终发送所有的承诺和折半响应为： $(L_1,R_1),\ldots,(L_k,R_k),(a,b)$。

数据长度为 $2k+\frac{2n}{2^k}=2k+2$。其中，$k=\log_2 n$。

最终校验等式为： $\left(L_1^{(x_1^2)}...L_k^{(x_k^2)}\right)\cdot P_1\cdot\left(R_k^{(x_k^{-2})}...R_1^{(x_1^{-2})}\right)==Hash\left(x_k^{-1}a,x_ka,x_kb,x_k^{-1}b,\langle a,b\rangle\right)$ 运算关系 $\boxed{k}$

3.2.2 指数版本向量内积承诺

Prover	Verifier
公共输入 $P,\vec{g},\vec{h},u$	
保密输入 2 个 n 维向量 $\vec{a},\vec{b}$ 。	
如果 $n=1$，则发送 a,b。	
	校验 $P==g^a h^b u^{a\cdot b}$。
如果 $n>1$，则计算折半 $n'=n/2$ 计算承诺 L 和 R： $c_L=\left\langle \vec{a}_{[:n']},\vec{b}_{[n':]}\right\rangle\in\mathbb{Z}_p$ $c_R=\left\langle \vec{a}_{[n':]},\vec{b}_{[:n']}\right\rangle\in\mathbb{Z}_p$ $L=\vec{g}_{[n':]}^{\vec{a}_{[:n']}}\vec{h}_{[:n']}^{\vec{a}_{[n':]}}u^{c_L}\in\mathbb{G}$ $R=\vec{g}_{[:n']}^{\vec{a}_{[n':]}}\vec{h}_{[n':]}^{\vec{a}_{[:n']}}u^{c_R}\in\mathbb{G}$ 计算挑战： $x=SHA3(P,L,R)$ 更新： $\vec{g}'=\vec{g}_{[:n']}^{x^{-1}}\circ\vec{g}_{[n':]}^{x}$ $\vec{h}'=\vec{h}_{[:n']}^{x^{-1}}\circ\vec{h}_{[n':]}^{x}$ $P'=L^{x^2}PR^{x^{-2}}$ 不断刷新 计算**折半响应**： $\vec{a}'=\vec{a}_{[:n']}\cdot x+\vec{a}_{[n':]}\cdot x^{-1}$ $\vec{b}'=\vec{b}_{[:n']}\cdot x+\vec{b}_{[n':]}\cdot x^{-1}$ **发送**承诺 L,R 和响应 $\vec{a}',\vec{b}'$	
	接收承诺 L,R 和响应 $\vec{a}',\vec{b}'$ 计算挑战： $x=SHA3(P,L,R)$ 更新：

	$\vec{g}\,' = \vec{g}_{[:n']}^{x^{-1}} \circ \vec{g}_{[n':]}^{x}$ $\vec{h}\,' = \vec{h}_{[:n']}^{x^{-1}} \circ \vec{h}_{[n':]}^{x}$ $P' = L^{x^2} P R^{x^{-2}}$ 分析：更新次数多，意味着计算复杂度较高，相比 KZG 承诺是缺点。

3.2.3 秘密提取

回滚1次

标准的 Sigma 协议 4 个步骤：承诺、挑战、响应、校验。
理论上，如果能（时间倒流）回滚到**挑战**步骤，则承诺不变，**挑战**和**响应**对应更新。

Sigma 协议：承诺 $A = g^r$、挑战 e_1、响应 $z_1 = r + e_1 \cdot \omega$、校验 $z_1 \cdot G = A + e_1 \cdot Q$。

回滚后：新挑战 e_2、新响应 $z_2 = r + e_2 \cdot \omega$、校验 $z_2 \cdot G = A + e_2 \cdot Q$。

验证方根据 2 个挑战 e_1, e_2 和 2 个响应 z_1, z_2，有 2 个方程：

$$z_1 = r + e_1 \cdot \omega$$
$$z_2 = r + e_2 \cdot \omega$$

能够计算秘密 ω，然后校验 $Q == \omega \cdot G$，**确保秘密确实存在且正确。**

因此，上述**向量内积的 Sigma 协议**在回滚后，也要能提取秘密向量 $\vec{a}, \vec{b}$，以确保秘密确实存在。

定理：如果基于向量内积承诺 Sigma 协议回滚 4 次，则能够提取离散对数或秘密向量 $\vec{a}, \vec{b}$。

该定理确保秘密确实存在且正确。

证明：

（1）如果 $n = 1$，　则 $P = g^a h^b u^{a \cdot b}$。

证明方发送 a, b，则验证方获得秘密 a, b，并检测 $P == g^a h^b u^{a \cdot b}$，并计算获得 $c = a \cdot b$。

（2）如果 $n = 2^k$。构造一个**提取图灵机** Ω_1，Ω_1 运行证明方 $\mathcal{P}$，获得承诺 L, R；

如果回滚 4 次，则获得 **4 个挑战** $x_i, i = 1,2,3,4$，且 $x_i \neq \pm x_j, 1 \le i < j \le 4$ 和 **4 个折半响应**

$(\vec{a}_i', \vec{b}_i'), i = 1,2,3,4$，满足运算关系

只折了一次

$$L_1^{(x_i^2)} \cdot P_1 \cdot R_1^{(x_i^{-2})} == \left(\vec{g}_{[:n']}^{x_i^{-1}} \circ \vec{g}_{[n':]}^{x_i} \right)^{\vec{a}_i'} \cdot \left(\vec{h}_{[:n']}^{x_i} \circ \vec{h}_{[n':]}^{x_i^{-1}} \right)^{\vec{b}_i'} \cdot u^{\langle \vec{a}_i', \vec{b}_i' \rangle}, i = 1,2,3,4 \text{ 公式(1)}$$

通过以下三个方程

$$x_1^2\eta_1 + x_2^2\eta_2 + x_3^2\eta_3 = 1$$
$$\eta_1 + \eta_2 + \eta_3 = 0$$
$$x_1^{-2}\eta_1 + x_2^{-2}\eta_2 + x_3^{-2}\eta_3 = 0$$

解方程组求解$\eta_1,\eta_2,\eta_3 \in \mathbb{Z}_p$。

基于η_1,η_2,η_3对**公式(1)中** i=1, 2, 3 进行线性组合

$$\left(L_1^{(x_1^2)} \cdot P_1 \cdot R_1^{(x_1^{-2})}\right)^{\eta_1} == \left(\vec{g}_{[:n']}^{x_1^{-1}} \circ \vec{g}_{[n':]}^{x_1}\right)^{\eta_1 \vec{a}_1'} \cdot \left(\vec{h}_{[:n']}^{x_1} \circ \vec{h}_{[n':]}^{x_1^{-1}}\right)^{\eta_1 \vec{b}_1'} \cdot u^{\eta_1\langle \vec{a}_1',\vec{b}_1'\rangle}$$
$$\left(L_1^{(x_2^2)} \cdot P_1 \cdot R_1^{(x_2^{-2})}\right)^{\eta_2} == \left(\vec{g}_{[:n']}^{x_2^{-1}} \circ \vec{g}_{[n':]}^{x_2}\right)^{\eta_2 \vec{a}_2'} \cdot \left(\vec{h}_{[:n']}^{x_2} \circ \vec{h}_{[n':]}^{x_2^{-1}}\right)^{\eta_2 \vec{b}_2'} \cdot u^{\eta_2\langle \vec{a}_2',\vec{b}_2'\rangle}$$
$$\left(L_1^{(x_3^2)} \cdot P_1 \cdot R_1^{(x_3^{-2})}\right)^{\eta_3} == \left(\vec{g}_{[:n']}^{x_3^{-1}} \circ \vec{g}_{[n':]}^{x_3}\right)^{\eta_3 \vec{a}_3'} \cdot \left(\vec{h}_{[:n']}^{x_3} \circ \vec{h}_{[n':]}^{x_3^{-1}}\right)^{\eta_3 \vec{b}_3'} \cdot u^{\eta_3\langle \vec{a}_3',\vec{b}_3'\rangle}$$
$$\left(L_1^{(\eta_1 x_1^2 + \eta_2 x_2^2 + \eta_3 x_3^2 = 1)} \cdot P_1^{\eta_1+\eta_2+\eta_3=0} \cdot R_1^{(\eta_1 x_1^{-2} + \eta_2 x_2^{-2} + \eta_3 x_3^{-2} = 0)}\right) = L_1 = \vec{g}^{a_L} \cdot \vec{h}^{b_L} \cdot u^{c_L}$$

则能够计算$\vec{a}_L,\vec{b}_L \in \mathbb{Z}_p^n, c_L \in \mathbb{Z}_p$满足运算关系：$L_1 = \vec{g}^{\vec{a}_L} \cdot \vec{h}^{\vec{b}_L} \cdot u^{c_L}$。

同理，通过另外 3 个方程

$$x_1^2\eta_1' + x_2^2\eta_2' + x_3^2\eta_3' = 0$$
$$\eta_1' + \eta_2' + \eta_3' = 1$$
$$x_1^{-2}\eta_1' + x_2^{-2}\eta_2' + x_3^{-2}\eta_3' = 0$$

解方程组求解$\eta_1',\eta_2',\eta_3' \in \mathbb{Z}_p$。对**公式(1)中** i=1, 2, 3 进行线性组合，则能够计算

$\vec{a}_P,\vec{b}_P \in \mathbb{Z}_p^n, c_P \in \mathbb{Z}_p$满足运算关系：$P_1 = \vec{g}^{\vec{a}_P} \cdot \vec{h}^{\vec{b}_P} \cdot u^{c_P}$。

再通过另外 3 个方程

$$x_1^2\eta_1'' + x_2^2\eta_2'' + x_3^2\eta_3'' = 0$$
$$\eta_1'' + \eta_2'' + \eta_3'' = 0$$
$$x_1^{-2}\eta_1'' + x_2^{-2}\eta_2'' + x_3^{-2}\eta_3'' = 1$$

解方程组求解$\eta_1'',\eta_2'',\eta_3'' \in \mathbb{Z}_p$。对**公式(1)中** i=1, 2, 3 进行线性组合，则能够计算

$\vec{a}_R,\vec{b}_R \in \mathbb{Z}_p^n, c_R \in \mathbb{Z}_p$满足运算关系：$R_1 = \vec{g}^{\vec{a}_R} \cdot \vec{h}^{\vec{b}_R} \cdot u^{c_R}$。

因此，**公式(1)**等价表达为

$$\vec{g}^{\vec{a}_L x^2 + \vec{a}_P + \vec{a}_R x^{-2}} \cdot \vec{h}^{\vec{b}_L x^2 + \vec{b}_P + \vec{b}_R x^{-2}} \cdot u^{c_L x^2 + c_P + c_R x^{-2}} = L_1^{x^2} P_1 R_1^{-x^2} = \vec{g}_{[:n']}^{\vec{a}' x^{-1}} \cdot \vec{g}_{[n':]}^{\vec{a}' x} \cdot \vec{h}_{[:n']}^{\vec{b}' x} \cdot \vec{h}_{[n':]}^{\vec{b}' x^{-1}} \cdot u^{\langle \vec{a}',\vec{b}'\rangle}$$

对于$x \in \{x_1,x_2,x_3,x_4\}$，上述各项对应相等，有以下等式

$$
\begin{aligned}
&(1):\vec{a}'x^{-1}=\vec{a}_{L,[:n']}x^{2}+\vec{a}_{P,[:n']}+\vec{a}_{R,[:n']}x^{-2}\\
&(2):\vec{a}'x=\vec{a}_{L,[n':]}x^{2}+\vec{a}_{P,[n':]}+\vec{a}_{R,[n':]}x^{-2}\\
&(3):\vec{b}'x=\vec{b}_{L,[n':]}x^{2}+\vec{b}_{P,[n':]}+\vec{b}_{R,[n':]}x^{-2}\\
&(4):\vec{b}'x^{-1}=\vec{b}_{L,[:n']}x^{2}+\vec{b}_{P,[:n']}+\vec{b}_{R,[:n']}x^{-2}\\
&(5):\left\langle \vec{a}',\vec{b}'\right\rangle=c_{L}x^{2}+c_{P}+c_{R}x^{-2}
\end{aligned}
$$

因为能计算出指数部分

如果上述 5 个等式不成立，则能够提取$(g_1,\ldots,g_n,h_1,\ldots,h_n)$之间的**离散对数**。

如果上述 5 个等式成立，则公式(1)(2)(3)(4)变为

$$
\begin{aligned}
&(1'):\vec{a}'=\vec{a}_{L,[:n']}x^{3}+\vec{a}_{P,[:n']}x+\vec{a}_{R,[:n']}x^{-1}\\
&(2'):\vec{a}'=\vec{a}_{L,[n':]}x+\vec{a}_{P,[n':]}x^{-1}+\vec{a}_{R,[n':]}x^{-3}\\
&(3'):\vec{b}'=\vec{b}_{L,[n':]}x+\vec{b}_{P,[n':]}x^{-1}+\vec{b}_{R,[n':]}x^{-3}\\
&(4'):\vec{b}'=\vec{b}_{L,[:n']}x^{3}+\vec{b}_{P,[:n']}x+\vec{b}_{R,[:n']}x^{-1}
\end{aligned}
$$

公式(1')=(2'), (3')=(4')，则有

$$
\begin{aligned}
&(5):\vec{a}_{L,[:n']}x^{3}+(\vec{a}_{P,[:n']}-\vec{a}_{L,[n':]})x+(\vec{a}_{R,[:n']}-\vec{a}_{P,[n':]})x^{-1}-\vec{a}_{R,[n':]}x^{-3}=0\\
&(6):\vec{b}_{L,[:n']}x^{3}+(\vec{b}_{P,[:n']}-\vec{b}_{L,[n':]})x+(\vec{b}_{R,[:n']}-\vec{b}_{P,[n':]})x^{-1}-\vec{b}_{R,[n':]}x^{-3}=0
\end{aligned}
$$

因为$x\in\{x_1,x_2,x_3,x_4\}$是 4 个随机数，所以等式(5)(6)成立的~~冲~~充要条件为

$$
\begin{aligned}
&\vec{a}_{L,[:n']}=\vec{a}_{R,[n':]}=\vec{b}_{L,[:n']}=\vec{b}_{R,[n':]}=0\\
&\vec{a}_{P,[:n']}=\vec{a}_{L,[n':]},\vec{a}_{R,[:n']}=\vec{a}_{P,[n':]},\\
&\vec{b}_{P,[:n']}=\vec{b}_{L,[n':]},\vec{b}_{R,[:n']}=\vec{b}_{P,[n':]}
\end{aligned}
$$

用于化简公式(2') (3')，得到

$$
\begin{aligned}
&(2'):\vec{a}'=\vec{a}_{P,[:n']}x+\vec{a}_{P,[n':]}x^{-1}\\
&(3'):\vec{b}'=\vec{b}_{P,[:n']}x+\vec{b}_{P,[n':]}x^{-1}
\end{aligned}
$$

带入公式(5)

$$
\begin{aligned}
&c_{L}x^{2}+c_{P}+c_{R}x^{-2}=\left\langle \vec{a}',\vec{b}'\right\rangle=\left\langle \vec{a}_{P,[:n']}x+\vec{a}_{P,[n':]}x^{-1},\vec{b}_{P,[:n']}x+\vec{b}_{P,[n':]}x^{-1}\right\rangle\\
&=\left\langle \vec{a}_{P,[:n']},\vec{b}_{P,[:n']}\right\rangle x^{2}+\left\langle \vec{a}_{P,[:n']},\vec{b}_{P,[n':]}\right\rangle+\left\langle \vec{a}_{P,[n':]},\vec{b}_{P,[:n']}\right\rangle+\left\langle \vec{a}_{P,[n':]},\vec{b}_{P,[n':]}\right\rangle x^{-2}\\
&=\left\langle \vec{a}_{P,[:n']},\vec{b}_{P,[:n']}\right\rangle x^{2}+\left\langle \vec{a}_{P},\vec{b}_{P}\right\rangle+\left\langle \vec{a}_{P,[n':]},\vec{b}_{P,[n':]}\right\rangle x^{-2}
\end{aligned}
$$

各项对应相等，因此**提取出**$c_P=\left\langle \vec{a}_P,\vec{b}_P\right\rangle$。

对于$n=2^k$，需要回滚$4^k=n^2$次，则能够计算出$c=\left\langle \vec{a},\vec{b}\right\rangle$。

构造**提取图灵机**Ω_2，提取图灵机Ω_1作为子协议，以获得秘密向量$\vec{a},\vec{b}$。

承诺为P，随机数x，秘密向量为$(\vec{a},\vec{b})$，满足运算关系

$$P\cdot u^{x\cdot c}==\vec{g}^{\vec{a}}\cdot\vec{h}^{\vec{b}}\cdot u^{x\cdot\langle\vec{a},\vec{b}\rangle}$$

回滚证明方Ω_1，新随机数x'和新秘密向量为$(\vec{a}',\vec{b}')$，满足运算关系

$$P\cdot u^{x'\cdot c}==\vec{g}^{\vec{a}'}\cdot\vec{h}^{\vec{b}'}\cdot u^{x'\cdot\langle\vec{a}',\vec{b}'\rangle}$$

上述两个等式相除，则有以下运算关系

$$u^{(x-x')\cdot c}==\vec{g}^{\vec{a}-\vec{a}'}\cdot\vec{h}^{\vec{b}-\vec{b}'}\cdot u^{x\cdot\langle\vec{a},\vec{b}\rangle-x'\cdot\langle\vec{a}',\vec{b}'\rangle}$$

- 如果$\vec{a}\neq\vec{a}',\vec{b}\neq\vec{b}'$，则计算出$\vec{g},\vec{h}$之间的**离散对数**为$\vec{a}-\vec{a}',\vec{b}-\vec{b}'$；
- 如果$\vec{a}=\vec{a}',\vec{b}=\vec{b}'$，则$u^{x\cdot c-x'\cdot c}==u^{x\cdot\langle\vec{a},\vec{b}\rangle-x'\cdot\langle\vec{a}',\vec{b}'\rangle}$，则计算出$c=\langle\vec{a},\vec{b}\rangle$，然后校验
  $c=\langle\vec{a},\vec{b}\rangle=\langle\vec{a}',\vec{b}'\rangle$
  $P==\vec{g}^{\vec{a}}\cdot\vec{h}^{\vec{b}}\cdot u^{a\cdot b}$，**确保秘密向量$(\vec{a},\vec{b})$确实存在且正确。**

3.3BulletProof 范围证明

- **Sigma 零知识证明：1 个承诺、1 个挑战、1 个响应和 1 个校验；**
- **BulletProof 范围证明：4 个承诺，3 个挑战，5 个响应和 3 个校验。**
  4 个承诺对应的秘密更多；3 个校验防止证明方作弊。

核心结论：以下 6 个运算关系①②③④⑤⑥等价。

分析：
- 运算关系①非常直观，但是不易证明；
- 运算关系⑥非常抽象，但是容易证明。

因此，如果证明知道秘密v满足运算关系⑥，则等价于证明知道秘密v满足运算关系①。

证明方证明知道金额秘密v和随机数γ，满足

$$\text{Pedersen承诺}\quad V=g^v h^\gamma,\quad v\in[0,2^n-1]\quad \text{运算关系①}$$

金额向量$\vec{a}_L=(a_1,...,a_n)\in\{0,1\}^n$，满足$\langle\vec{a}_L,2^n\rangle=v$。
v的二进制表达

门罗币中$n=2^{32}$，确保金额范围是$[0,2^{32}-1]$。
纯1

证明方基于金额向量$\vec{a}_L$计算向量承诺$A=h^\alpha\vec{g}^{\vec{a}_L}\vec{h}^{\vec{a}_R}\in\mathbb{G}$，且证明知道秘密$v,\gamma$满足

关系④与关系⑤等价，推导过程

$$\langle \vec{a}_L - z\cdot\vec{1}^n, \vec{y}^n \circ(\vec{a}_R + z\cdot\vec{1}^n) + z^2\vec{2}^n \rangle$$
$$= z^2\cdot v - z\langle \vec{a}_L - \vec{1}^n - \vec{a}_R, \vec{y}^n\rangle + \langle \vec{a}_L, z\cdot\vec{y}^n\rangle - z^3\langle\vec{1}^n,\vec{2}^n\rangle - z^2\langle\vec{1}^n,\vec{y}^n\rangle - z\langle\vec{1}^n,\vec{y}^n\circ\vec{a}_R\rangle$$
$$= z^2\cdot v - z^3\langle\vec{1}^n,\vec{2}^n\rangle - z^2\langle\vec{1}^n,\vec{y}^n\rangle + z\langle\vec{1}^n,\vec{y}^n\rangle + z\langle\vec{a}_R,\vec{y}^n\rangle - z\langle\vec{1}^n,\vec{y}^n\circ\vec{a}_R\rangle$$
$$= z\langle\vec{a}_R,\vec{y}^n\rangle - z\langle\vec{a}_R,\vec{a}_R\circ\vec{y}^n\rangle$$
$$= z\langle\vec{a}_R,\vec{y}^n\circ\vec{a}_L\rangle$$
$$=0$$

$$V = g^v h^\gamma,$$
$$\left\langle \vec{a}_L, \vec{2}^n \right\rangle = v, \underbrace{\vec{a}_L \circ \vec{a}_R = \vec{0}^n}_{\vec{a}_L\text{和}\vec{a}_R\text{正交}}, \vec{a}_R = \vec{a}_L - \vec{1}^n$$

运算关系②

作用：第 1 个 Pedersen 承诺用于金额绑定与隐藏；第 2 个 v 二进制展开为金额向量$\vec{a}_L$；第 3 个向量正交；第 4 个确保向量$\vec{a}_L, \vec{a}_R$为 0 或 1，是二进制表达，不能是其他进制表达。如果是其他进制表达，则范围空间增大。

选择随机数 $y \in \mathbb{Z}_p$，运算关系②用内积表达

内积放大 $\vec{y}^n$ $=\vec{a}_R$

$$V = g^v h^\gamma,$$
$$\left\langle \vec{a}_L, \vec{2}^n \right\rangle = v, \left\langle \vec{a}_L, \vec{a}_R \circ \vec{y}^n \right\rangle = 0, \left\langle \vec{a}_L - \vec{1}^n - \vec{a}_R, \vec{y}^n \right\rangle = 0$$

运算关系③

作用：第 3 个仍是正交关系。第 4 个确保$\vec{a}_L - \vec{1}^n - \vec{a}_R = 0 \Leftrightarrow \vec{a}_L = \vec{1}^n + \vec{a}_R$，确保是二进制。

后三个等式组合为一个内积运算：选择随机数 $z \in \mathbb{Z}_p$

$$V = g^v h^\gamma,$$
$$z^2 \left\langle \vec{a}_L, \vec{2}^n \right\rangle + z\left\langle \vec{a}_L - \vec{1}^n - \vec{a}_R, \vec{y}^n \right\rangle + \left\langle \vec{a}_L, \vec{a}_R \circ \vec{y}^n \right\rangle = z^2 \cdot v$$

运算关系④

用 Hadamard 乘积与内积运算等价表达

$$V = g^v h^\gamma,$$
$$\left\langle \vec{a}_L - z\cdot\vec{1}^n, \vec{y}^n \circ (\vec{a}_R + z\cdot\vec{1}^n) + z^2\vec{2}^n \right\rangle = z^2\cdot v + \delta(y,z)$$

运算关系⑤

其中，$\delta(y,z) = (z - z^2)\left\langle \vec{1}^n, \vec{y}^n \right\rangle - z^3\left\langle \vec{1}^n, \vec{2}^n \right\rangle \in \mathbb{Z}_p$。验证方也能够计算$\delta(y,z)$。

证明方：知道秘密为v,γ
计算金额向量 $\begin{aligned}&\vec{a}_L = \{0,1\}^n, such\_that \left\langle \vec{a}_L, \vec{2}^n \right\rangle = v,\\ &\vec{a}_R = \vec{a}_L - 1 \in \mathbb{Z}_p^n\end{aligned}$ （金额 V 二进制展开） 选择随机数$\alpha \in \mathbb{Z}_p$，计算**金额向量**$\vec{a}_L$ **承诺** $A = h^\alpha \vec{g}^{\vec{a}_L} \vec{h}^{\vec{a}_R} \in \mathbb{G}$。 选择随机向量$\vec{s}_L, \vec{s}_R \in \mathbb{Z}_p^n$，随机数$\rho \in \mathbb{Z}_p$，计算**随机向量的承诺** $S = h^\rho \vec{g}^{\vec{s}_L} \vec{h}^{\vec{s}_R} \in \mathbb{G}$。 **发送 2 个承诺** A, S；
计算 2 个挑战 $y, z = SHA256(V, g, h, A, S, i), i = 1, 2$；
金额向量$\vec{a}_L, \vec{a}_R$和随机向量$\vec{s}_L, \vec{s}_R$构造**多项式** $l(X) = (\vec{a}_L - z\cdot\vec{1}^n) + \vec{s}_L\cdot X$ $r(X) = \vec{y}^n \circ (\vec{a}_R + z\cdot\vec{1}^n + \vec{s}_R\cdot X) + z^2\vec{2}^n$

则 $t(X)=\langle l(X),r(X)\rangle=t_0+t_1X+t_2X^2$

其中

$$V=g^{v}h^{\gamma},$$
$$t_0==z^2\cdot v+\delta(y,z)$$
运算关系⑥

证明方**证明**知道金额向量 $\vec{a}_L,\vec{a}_R$ 和随机向量 $\vec{s}_L,\vec{s}_R$ 满足运算关系⑥，则等价于证明知道金额向量 $\vec{a}_L,\vec{a}_R$ 和随机向量 $\vec{s}_L,\vec{s}_R$ 满足运算关系⑤④③②① $v\in[0,2^n-1]$。

选择随机数 $\tau_1,\tau_2\in\mathbb{Z}_p$，计算 **2 个多项式系数** t_1,t_2 **的承诺** $T_1=g^{t_1}h^{\tau_1},T_2=g^{t_2}h^{\tau_2}$。

发送 2 个承诺 T_1,T_2；

计算 1 个挑战 $x=SHA256(V,g,h,A,S,T_1,T_2)$。

随机向量 $\vec{s}_L,\vec{s}_R$ 对金额向量 $\vec{a}_L,\vec{a}_R$ 起随机化作用。

基于秘密金额向量 $\vec{a}_L,\vec{a}_R$、随机向量 $\vec{s}_L,\vec{s}_R$、随机数 τ_1,τ_2，计算 **5 个响应**

$$\vec{l}=l(x)=\vec{a}_L-z\cdot\vec{1}^n+\vec{s}_L\cdot x\in\mathbb{Z}_p^n$$
$$\vec{r}=r(x)=\vec{y}^n\circ(\vec{a}_R+z\cdot\vec{1}^n+\vec{s}_R\cdot x)+z^2\vec{2}^n\in\mathbb{Z}_p^n$$
$$\hat{t}=\langle\vec{l},\vec{r}\rangle\in\mathbb{Z}_p$$
$$\tau_x=\tau_2\cdot x^2+\tau_1\cdot x+z^2\gamma$$
$$\mu=\alpha+\rho x\in\mathbb{Z}_p^n$$

发送 5 个响应 $\tau_x,\mu,\hat{t},\vec{l},\vec{r}$。 优化

验证方：

计算 $h_i'=h_i^{(y^{-i+1})}\in\mathbb{G}$，构造向量 $\vec{h}'=(h_1,\ldots,h_n)$；

计算承诺 $P=A\cdot S^x\cdot\vec{g}^{-z}\cdot(\vec{h}')^{z\cdot\vec{y}^n+z^2\cdot\vec{2}^n}$；

3 个校验：

$$g^{\hat{t}}\cdot h^{\tau_x}==V^{z^2}\cdot g^{\delta(y,z)}\cdot T_1^x\cdot T_2^{x^2}$$
$$P==h^{\mu}\cdot\vec{g}^{\vec{l}}\cdot(\vec{h}')^{\vec{r}}$$
可优化，折半响应
$$\hat{t}==\langle\vec{l},\vec{r}\rangle$$

分析：

校验公式 1 公式推导：

$$
\begin{aligned}
V^{z^2} \cdot g^{\delta(y,z)} \cdot T_1^x \cdot T_2^{x^2} &= (g^v h^\gamma)^{z^2} g^{\delta(y,z)} (g^{xt_1} h^{x\tau_1})(g^{x^2 t_2} h^{x^2\tau_2}) \\
&= g^{vz^2 + xt_1 + x^2 t_2 + \delta(y,z)} h^{\gamma z^2 + x\tau_1 + x^2\tau_2} = g^{t_0 + xt_1 + x^2 t_2} h^{\gamma z^2 + x\tau_1 + x^2\tau_2} \\
&= g^{\hat{t}} \cdot h^{\tau_2 \cdot x^2 + \tau_1 \cdot x + z^2 \gamma}
\end{aligned}
$$

确保$\hat{t} = t_0 + t_1 x + t_2 x^2$，从而确保运算关系⑥正确。

校验公式 2 公式推导：

$$
\begin{aligned}
A \cdot S^x \cdot \vec{g}^{-z} \cdot (\vec{h}')^{z\cdot\vec{y}^n + z^2\cdot\vec{2}^n} &= (h^\alpha \vec{g}^{\vec{a}_L} \vec{h}^{\vec{a}_R})(h^\rho \vec{g}^{\vec{s}_L} \vec{h}^{\vec{s}_R})^x \cdot \vec{g}^{-z} \cdot (\vec{h}')^{z\cdot\vec{y}^n + z^2\cdot\vec{2}^n} \\
h^\mu \cdot \vec{g}^{\vec{l}} \cdot (\vec{h}')^{\vec{r}} &= h^{\alpha+\rho x} \cdot \vec{g}^{\vec{a}_L - z\cdot\vec{1}^n + \vec{s}_L \cdot x} \cdot (\vec{h}')^{\vec{y}^n \circ (\vec{a}_R + z\cdot\vec{1}^n + \vec{s}_R \cdot x) + z^2 \vec{2}^n}
\end{aligned}
$$

确保**响应向量**$\vec{l}, \vec{r}$包含**金额向量**$\vec{a}_L, \vec{a}_R$。

校验公式 3 作用：确保$\hat{t}$是基于$\vec{l}, \vec{r}$计算的。

上述 3 个校验，则能够防止证明方作恶，**确保金额范围**$v \in [0, 2^n - 1]$。

优化：响应$\vec{l}, \vec{r}$是 n 维向量，满足运算关系$P == h^\mu \cdot \vec{g}^{\vec{l}} \cdot (\vec{h}')^{\vec{r}}$，发送数据为$2n$。

优化：使用**向量内积承诺，折半响应**，修改为发送$(L_1, R_1), \ldots, (L_k, R_k), (a, b)$，长度为$2k+2$，其中$k = \log_2 n$。

3.4 批量范围证明

证明方知道m个秘密v_j, γ_j，满足运算关系

$$
\begin{aligned}
&V_j = g^{v_j} h^{\gamma_j}, \\
&v_j \in [0, 2^n - 1], j = 1, \ldots, m
\end{aligned}
\quad ①
$$

需要对应**修改部分：**

$\vec{a}_L = \{0,1\}^{n\cdot m}, such\_that \left\langle \vec{a}_L[(jn-n : jn-1], \vec{2}^n \right\rangle = v_j,$

$\vec{a}_R = \vec{a}_L - \overset{\rightarrow n\cdot m}{1} \in \mathbb{Z}_p^{n\cdot m}$

$l(X) = (\vec{a}_L - z\cdot\vec{1}^{n\cdot m}) + \vec{s}_L \cdot X \in \mathbb{Z}_p^{n\cdot m}[X]$

$r(X) = y^{n\cdot m} \circ (\vec{a}_R + z\cdot\vec{1}^{n\cdot m} + \vec{s}_R \cdot X) + \sum_{j=1}^{m} z^{1+j} (\vec{0}^{jn-n} \,||\, 2^n \,||\, \vec{0}^{mn-jn})$

$\tau_x = \tau_1 x + \tau_2 x^2 + \sum_{j=1}^{m} (z^{1+j} \gamma_j)$

$\delta(y,z) = (z - z^2)\left\langle 1^{n\cdot m}, y^{n\cdot m} \right\rangle - \sum_{j=1}^{m} \left(z^{1+j} \left\langle \vec{1}^n, \vec{2}^n \right\rangle \right)$

$$g^{\hat{t}} \cdot h^{\tau_x} == \vec{V}^{z^2 \cdot \vec{z}^m} \cdot g^{\delta(y,z)} \cdot T_1^x \cdot T_2^{x^2}, \vec{V} = (V_1, ..., V_m)$$

$$P = A \cdot S^x \cdot g^{-z} \cdot (\vec{h}')^{z \cdot y^{n \cdot m}} \prod_{j=1}^{m} (\vec{h}')_{jn-n:jn-1}^{z^{j+1} \cdot 2^n}$$

4.Diffie-Hellman 密钥交换

4.1Diffie-Hellman 密钥交换

Alice	Bob
私钥 $SK_1 = \alpha$，公钥 $PK_1 = g^\alpha$	私钥 $SK_2 = \beta$，公钥 $PK_2 = g^\beta$
发送公钥 PK_1	发送公钥 PK_2
计算 $(PK_2)^{SK_1} = (g^\beta)^\alpha = g^{\alpha\beta}$	计算 $(PK_1)^{SK_2} = (g^\alpha)^\beta = g^{\alpha\beta}$
$(PK_2)^{SK_1} = (g^\beta)^\alpha = g^{\alpha\beta} = (g^\alpha)^\beta = (PK_1)^{SK_2}$ **会话密钥或公共密钥** $key = g^{\alpha\beta}$，或 $key = Hash\left(g^{\alpha\beta}\right)$	

有 2 个缺点：
缺点 1：公共密钥永远没变化。
改进：添加公开随机数 r

4.2 添加随机数

Alice	Bob
私钥 $SK_1 = \alpha$，公钥 $PK_1 = g^\alpha$	私钥 $SK_2 = \beta$，公钥 $PK_2 = g^\beta$
发送公钥 PK_1 和随机数 r_1	发送公钥 PK_2 和随机数 r_2
计算 $(PK_2)^{SK_1} = (g^\beta)^\alpha = g^{\alpha\beta}$	计算 $(PK_1)^{SK_2} = (g^\alpha)^\beta = g^{\alpha\beta}$
会话密钥或公共密钥 $key = Hash\left(g^{\alpha\beta}, r_1, r_2\right)$	

因此 Alice 与 Bob 计算出相同的会话密钥 Key。**会话密钥每次都会发生变化！！！**

4.3 中间人攻击

缺点 2：中间人攻击

Alice	**Adversary**	Bob
私钥 $SK_1=\alpha$， 公钥 $PK_1=g^{\alpha}$	私钥 $SK_A=\omega$， 公钥 $PK_A=g^{\omega}$	私钥 $SK_2=\beta$， 公钥 $PK_2=g^{\beta}$
发送公钥 PK_1		
	接收公钥 PK_1 中间人获取 g^{α} 修改为 **发送**公钥 PK_A **给双方**	
接收公钥 PK_A		**接收**公钥 PK_A **发送**公钥 PK_2
	接收公钥 PK_2 获取 g^{β}	
计算会话密钥 $key=Hash\left(g^{\alpha\omega}\right)$ 中间人也知道		
	计算会话密钥 $key=Hash\left(g^{\beta\omega}\right)$	

4.4 添加随机数不能解决中间人攻击

Alice	**Adversary**	Bob
私钥 $SK_1=\alpha$， 公钥 $PK_1=g^{\alpha}$	私钥 $SK_A=\omega$， 公钥 $PK_A=g^{\omega}$	私钥 $SK_2=\beta$， 公钥 $PK_2=g^{\beta}$
发送公钥 PK_1 和随机数 r_1		
	接收公钥 PK_1 和随机数 r_1 修改为 **发送**公钥 PK_A 和随机数 r_A **给双方**	
接收公钥 PK_A 和随机数 r_A		**接收**公钥 PK_A 和随机数 r_A **发送**公钥 PK_2 和随机数 r_2
	接收公钥 PK_2 和随机数 r_2	

计算会话密钥 $key = Hash\left(g^{\alpha\omega}, r_1, r_A\right)$		
	计算会话密钥 $key = Hash\left(g^{\beta\omega}, r_A, r_2\right)$	

公钥证书（双方认证）能够解决中间人攻击。**攻击者只能截断，不能窃听。**但是，**证书**是**中心化**的系统有拒绝服务攻击和反应迟缓等问题。

4.5 三方 Diffie-Hellman 密钥协商协议

Alice	Bob	Carol
私钥 $SK_1 = \alpha$， 公钥 $PK_1 = g^{\alpha}$	私钥 $SK_2 = \beta$， 公钥 $PK_2 = g^{\beta}$	私钥 $SK_3 = \chi$， 公钥 $PK_3 = g^{\chi}$
计算会话密钥 $k_1 = Hash\left(g^{\alpha\beta}\right)$ 计算对应的公共公钥 $K_1 = g^{k_1}$ 发送公共公钥 K_1		发送公钥 PK_3
$k_2 = Hash\left(g^{k_1\chi}\right)$		

如果再加一个参与方 Dave，则 Alice, Bob,Carol 与 Dave 计算共同的会话密钥 Key_{ψ} 对于的公钥 $PK_{\psi} = g^{Key_{\psi}}$ 。

四个参与方一起计算共享会话密钥，直到 n 个参与方共享会话密钥。
用共享的会话密钥和 **GCM-AES** 加密加密数据。
分析：如果连续的，每次加入一个参与方，则协商一次，复杂度呈线性增加。
如果同时加入多个参与方，则各个参与方两两协商，形成二叉树结构。