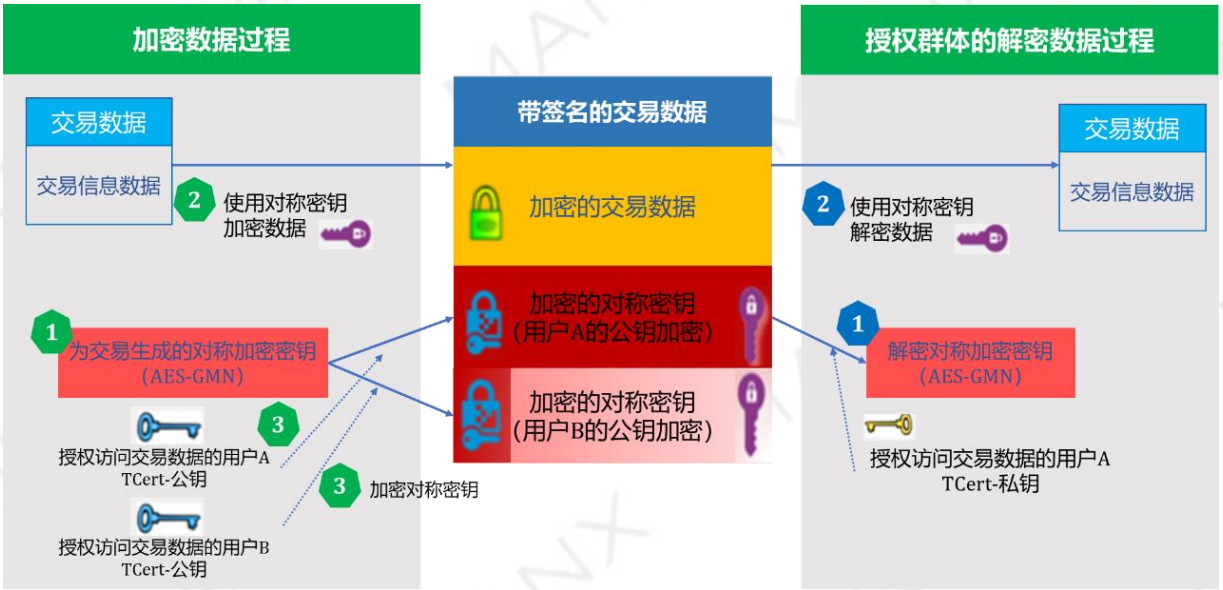
# MANX 抗量子加密技术

## 课题 1：加密模块是数据授权共享的最基础技术

数据授权共享通过对称密码和非对称秘钥联合实施。具体步骤如下：

1) 数据共享的发起节点为交易生成对称加密秘钥；
2) 使用对称加密秘钥加密上链的交易数据和其他数据，得到密文信息；
3) 选择数据授权对象或者对象集合，并查询授权对象的公钥信息；
4) 通过授权对象的公钥信息加密对称加密秘钥，并发送给授权共享对象；
5) 授权对象通过自己的私钥解密得到对称加密秘钥；
6) 授权对象通过解密秘钥去对应的存储路径找到相应的密文信息，并解密得到授权共享的原始数据。




*数据授权共享*

## 课题 2：MANX 加密模块的通用性

MANX 加密模块实现可插拔加解密模块可随时支持国际加密标准 AES、国标密码体制 SM4 的无缝切换；签名与验证模块支持椭圆曲线 ECC、国标 SM2 签名算法和验证算法，因此，可插拔加密模

块可以在遵循相同加密协议的所有情况下使用；区块链底层协议协处理模块支持现有通用公有链平台的交易打包、发送和验证。

### 课题 **3**：量子计算对传统加密算法的威胁

近三十年，以 RSA、ECC、Diffie-Hellman、代数同态为代表的公钥密码体制为代表的加密方法已成为互联网与信息基础设施的核心安全协议，在军事、政治、 经济、生活等方面都有广泛且关键的应用，在个人、企业和政府的安全通讯中发挥着至关重要的作用。量子计算对于传统密码的理论威胁、量子计算机相关技术的高速发展、各国政府量子战略和政策的启动和推进，无不让当今信息社会感受到一种前所未有的紧迫感。IBM 和 Google 已经分别研制成功 50 和 72 量子比特的量子计算原型机，将量子计算从理论向应用迈出了第一步。而以量子计算机为运行环境，量子算法能够轻易破解上述安全系统。

RSA 和 ECC 公钥体系的安全性建立在大整数分解和离散对数问题是 NP 难题，而 Shor 的量子算法在量子计算机环境下可以轻易破解上述 NP 难题。随着量子计算能力的快速成长，量子计算已经严重威胁着国家、金融、社会、个人等领域曾经牢不可破的安全防线，对几乎所有涉及信息安全的领域构成了巨大的威胁。在量子计算和抗量子算法方面，MANX 项目核心开发成员 Jack Chiu 博士和国际顾问 Professor Vladan Vuletic 有深入的交流。

区块链几乎将所有的数字货币的安全托付于公钥安全体系—椭圆曲线公钥密码 ECC，然而量子计算机走向应用后在几秒内即能破解包括比特币、以太坊在内的几乎所有的现行数字货币，使得数字货币的价值所在瞬间崩塌，因此我们提出了抗量子计算的密码安全体系及软件的解决方案，保障下一代数字货币的后量子时代的基础安全。

### 课题 **4**：**MANX** 抗量子安全体系建设

对抗量子密码体制的研究，在四个方面展开：

（1） 基于格的密码
（2） 基于编码的密码
（3） 多变量公钥密码
（4） 基于 Hash 函数

考虑到公钥和密文的规模可控与签名效率，我们将面向不同的生态和应用创建基于 HASH 函数和格密码的后量子公钥加密和签名标准。


*抗量子安全体系*

课题 **5**：**MANX** 基于格密码的签名方案

格密码的安全性基于格问题中的最短向量和最近向量问题。学术界已经严格证明现有量子算法 Shor 和 Grover 无法解决上述难题，所以区别于 RSA 和 ECC 容易被量子攻击的情况，格密码对于经典和量子攻击均能免疫，其核心原因在于格密码的安全基石对于任意算法的抵抗性。

1）参数选取

从$\{-d,\dots,0,\dots,d\}$中随机选取整数构造$m \times k$维矩阵$S$，作为私钥，也就是签名秘钥。同样的，在$Z_q$上随机选取整数构造n × m矩阵 A，计算$T = AS \in Z_q^{m\times k}$，作为公钥，也就是验证秘钥。假定格签名方案中的哈希函数$H:\{0,1\}^* \to \{v: v \in \{-1,0,1\}^k, \|v\|_1 \le \kappa\}$。为了对消息$\mu$进行签名，签名首先在离散均匀分布$D_\sigma^m$中随机选取$m$维向量 y，其中，σ是$D_\sigma^m$的标准差，即得到格签名方案的相关参数。

2）秘钥生成

签名秘钥：从$\{-d,\dots,0,\dots,d\}$随机选取$S$；

验证秘钥：从$Z_q^{n\times m}$中随机选取 A，计算$T = AS \in Z_q^{m\times k}$；

随机预言机模型：

$$H:\{0,1\}^* \to \{v: v \in \{-1,0,1\}^k, \|v\|_1 \le \kappa\}$$

3）签名过程

步骤一：自离散均匀分布$D_\sigma^m$中随机选取$m$维向量 y；

步骤二：计算$Ay$，然后计算$c = H(Ay, \mu)$，其中$\mu$为签名消息；

步骤三：计算$z = Sc + y$；

步骤四：依概率$\Pr = \min\left(\frac{D_\sigma^m(z)}{MD_{Sc,\sigma}^m(z)}, 1\right)$，输出签名结果(z, c)，其中签名过程采用了拒绝采样定理。

4）验证过程

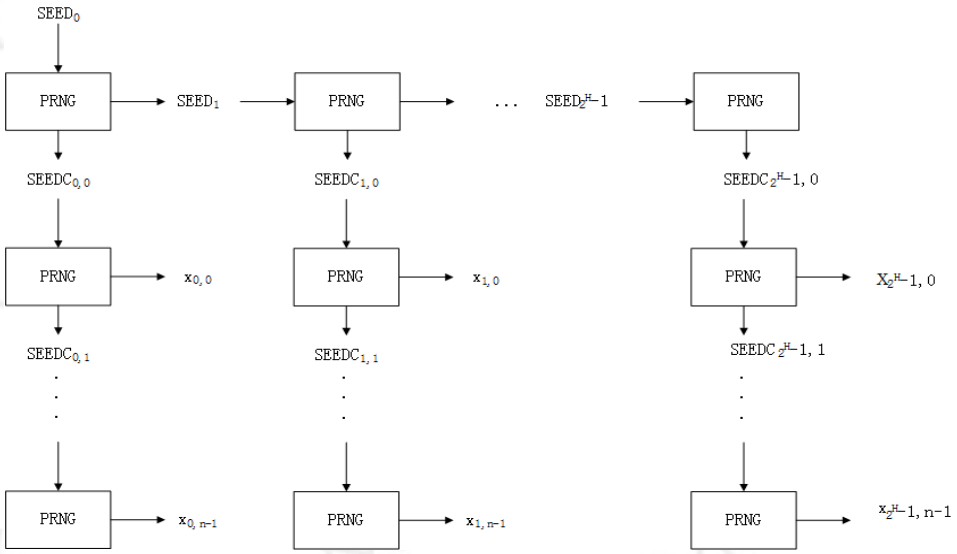
收到签名对(z, c)后计算$\|z\|$，如果满足$\|z\| \le 2\sigma\sqrt{m}$且$c = H(Az - Tc, \mu)$，则接受签名。

## 课题 6：MANX 基于默克尔树的签名方案

步骤一：通过 PN256 生成模块生成 HASH 函数的随机参数；

步骤二：通过伪随机数生成器 PRNG，使用种子生成签名与验证秘钥序列；

PRNG：$\{0,1\}^n \to \{0,1\}^n \times \{0,1\}^n$




*伪随机数生成器*

步骤三：使用签名与验证秘钥序列生成默克尔树的根节点值，作为签名系统的公钥；

$v_{0,j} = \text{hash}(Y_j)$, $v_{h,j} = \text{hash}(v_{h-1,2j} \| v_{h-1,2j+1})$，$1 \le h \le H$，$0 \le j < 2^{H-h}$

步骤四：查询计数器模块，得到本次签名秘钥的坐标值；

步骤五：根据本次签名秘钥的坐标值，计算用于本次签名的签名秘钥、验证秘钥和默克尔树验证路径的节点值；

步骤六：通过消息摘要生成模块处理待签名消息，生成消息摘要；

步骤七：利用步骤五产生的签名秘钥、验证秘钥和默克尔树验证路径的节点值，对步骤六的消息摘要进行一次性签名：

$$e_s = (e_{s,n-1}, e_{s,n-2}, \dots, e_{s,0}) = \left(f^{d_{n-1}}(x_{s,n-1}), f^{d_{n-2}}(x_{s,n-1}), \dots, f^{d_1}(x_{s,1}), f^{d_0}(x_{s,0})\right)$$

步骤八：计数器模块加 1，指向下一次的签名秘钥。

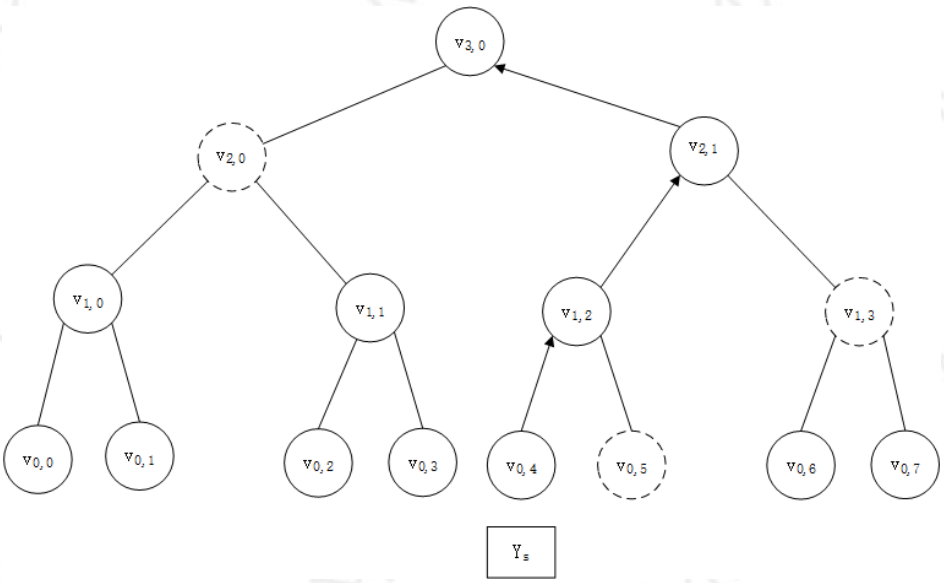
其中验证实现方法包括以下几个步骤：

步骤一：通过消息摘要生成模块处理待签名消息，生成消息摘要；

步骤二：通过一次性签名验证模块验证消息摘要的一次性签名部分的有效性；

步骤三：通过默克尔树根节点验证模块判断验证秘钥的有效性；

$T_0=\text{hash}(Y_s)$，若$\lfloor s/2^h \rfloor = 0 \bmod 2$，则 $T_h= \text{hash}(T_{h-1}|| p_{h-1})$，若$\lfloor s/2^h \rfloor = 1 \bmod 2$，则 $T_h= \text{hash}(p_{h-1} ||T_{h-1})$，这里 $1 \le h \le H$。




*默克尔树根节点验证秘钥的有效性*

步骤四：通过综合判断模块判断本次签名的整体有效性。

| | MANX（lattice） | MANX（Hash） | InterValue | Hcash |
|---|---|---|---|---|
| 原理 | ETRUsystem 签名和加密体系 | Chain Merkle 签名（CMSS） | NTRUsign 签名算法 | LWE 签名算法 |
| 签名速率 | 最快 | 较快 | 较快 | 中 |
| 密钥长度 | 较小 | 最小 | 中等 | 中等 |
| 签名长度 | 最小 | 较小 | 较小 | 中等 |
| 加密功能和点对点定向共享 | 支持 | 不支持 | 不支持 | 不支持 |
| 同态计算 | 支持 | 不支持 | 不支持 | 不支持 |
| | 适用于高速场景 | 适用于秘钥长度可控的场景 | | |

*MANX 综合判断模块*