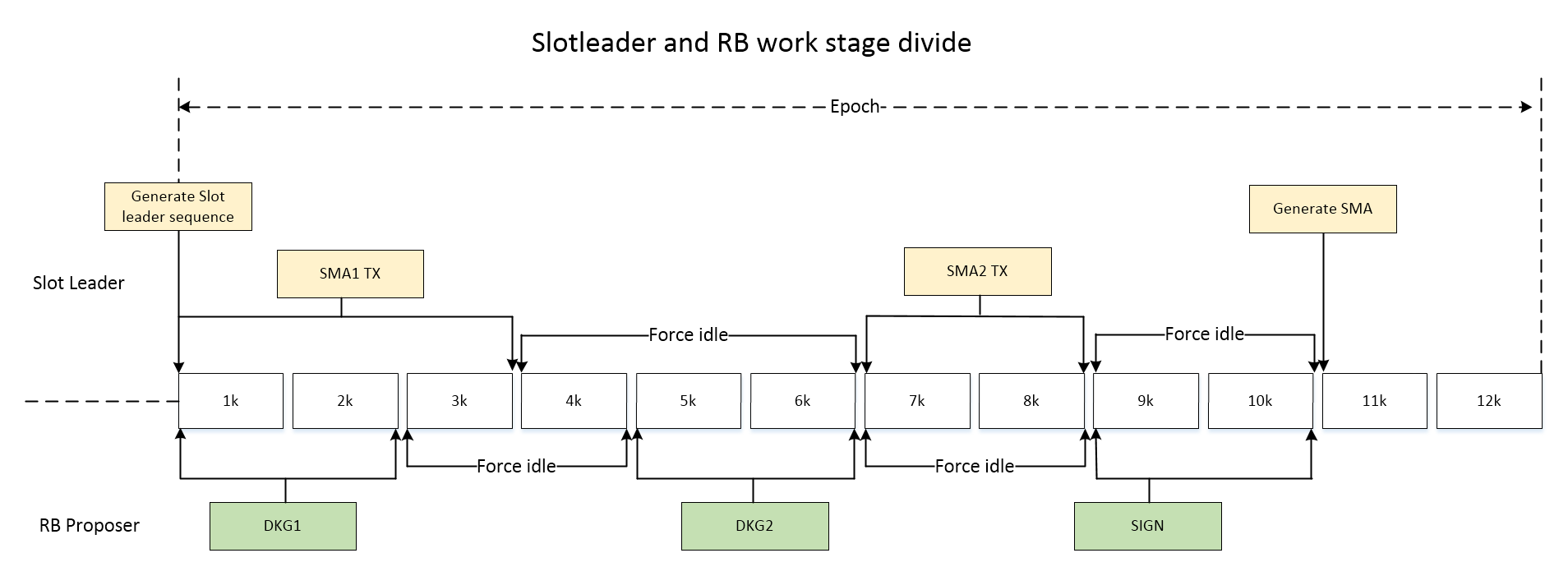
## Random Beacon的预编译合约

**RB协议：**Random Beacon协议（简称RBP），主要解决随机数的生成问题。在每个epoch开始，会选举出一个**随机数生成小组**(Random Beacon Group简称RBG)，这个组中的各个成员会在该epoch期间，执行协议，将交流信息上链，最终算出下个epoch将要使用的随机数。

**RB预编译合约(简称RBC)：**交流信息上链，其实就是执行预编译合约的过程。它主要负责检查信息是否正确，将必要的信息保存，以及算出随机数。



Epoch包含12 k个slot。每个slot对应一段等长的时间，比如10秒

合约有三个核心方法：

**dkg1：** dkg全称是decentralized key generation。此方法在epoch的前2k个slot才能顺利执行，最终把合法的commit信息保存到预编译合约的状态树里

**dkg2：** 此方法在5k、6k这2k个slot期间才能顺利执行，最终把encrypt share信息保存到预编译合约的状态树里

**sigShare：**是signature of its public key share。此方法在9k、10k这2k个slot期间才能顺利执行，最终计算出下个epoch需要的随机数。

### 核心方法处理过程详解

每个核心方法都进行的normalCheck检查有：

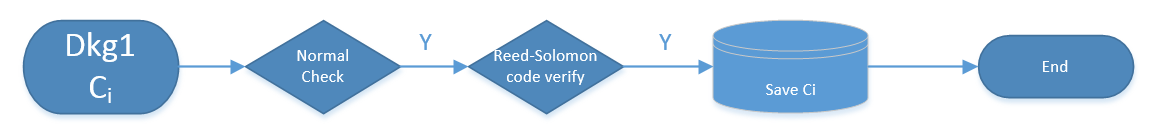
1. checkTime: 检查交易是否发生在相应的epoch，且在epoch的在对应的slot期间内，
2. checkProposer:检查交易发起人是否在该epoch的**随机数生成小组**内
3. checkSize: 检查数据项的长度，是否和**随机数生成小组**的大小相等
4. checkJoined：检查是否已经参与过

设随机数生成小组的顺序为P1,P2,...,Pn，他们的公钥为PK1, PK2,...,PKn，PKi之间有可能重复

### dkg1

本阶段核心检查和存储：

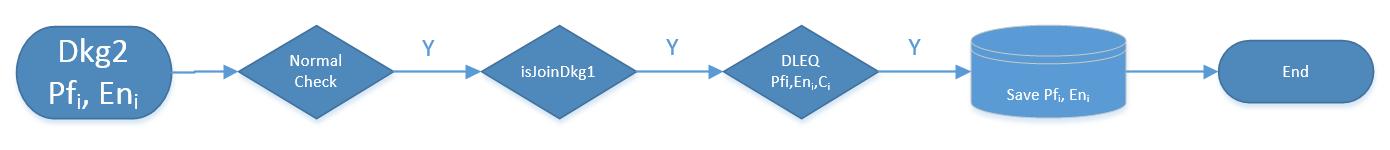
1. 计算X
   1. 先计算PK1+1，PK2+2，……，PKn+n，
   2. 再做hash, 设H=crypto.Keccak256得到H(PK1+1)，H(PK2+1)，……，H(PKn+1),
   3. 转成大整数，对bn256曲线阶取模xi=H(PKi+i) mod bn256.P
2. 做Reed-Solomon code 验证，验证参数Ci是否和垂直空间中的向量正交，如果验证通过，则存储提交的commit数据，记为Ci



### dkg2

本阶段核心检查和存储：

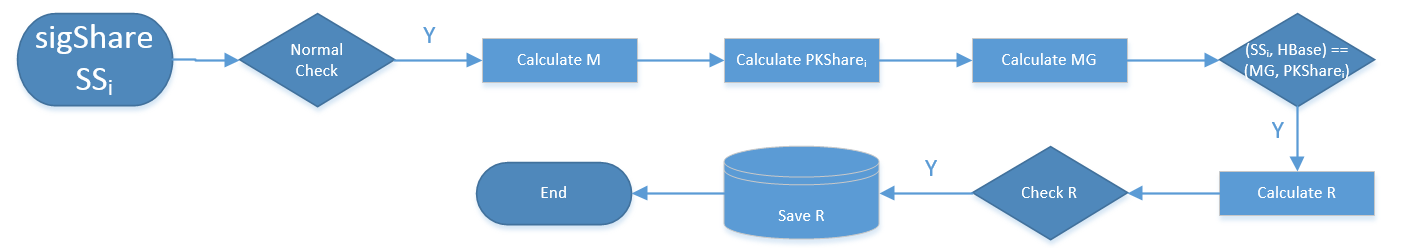
1. 是否参加dkg1——只有参加过dkg1的才能参与此阶段
2. dleq验证
   1. 通过参数解析出Proofi, EnSharei
   2. 对Proofi, EnSharei,Commiti做dleq验证，如果验证通过，则保存Ensharei



### sigShare

本阶段核心检查和存储

1. 计算M
   1. 设当前epoch的id为eid，则下个epoch的id为eid + 1
   2. 获取当前epoch的随机数ReId
   3. M = Hash(Reid + (eid + 1))
2. 计算gPKSharei
   1. 获取所有参与dkg2的人提交的Cj, 对j=1，2，……，n，累加第i个分量即为gPKSharei
3. 计算MG
   1. 如果参与sigShare的人数大于阈值，则计算MG=M倍乘bn256.G1
4. 验证bn256点是否相等，hBase为bn256曲线的基值
   1. 点pair1（参数gSignSharei,hBase）
   2. 点pair2（mG, gPKSharei）
   3. 如果pair1等于pair2，存储gSignSharei
5. 计算Reid+1
   1. 如果有效gSignSharei数量超过阈值，则开始计算Reid+1
   2. 由所有的gSignSharei，用基于G1的插值算出gSignature，则Reid+1=Hash(gSignature)
6. 验证Reid+1
   1. 计算X，过程和dkg1中的计算X相同
   2. 计算所有的gPKSharei，其中每一项和过程B一样，
   3. 由所有的gPKSharei，用基于G2的插值算出gPKShare
   4. 计算mG，同过程C
   5. 验证bn256点是否相等
      1. 点pair1(gSignature, hBase)
      2. 点pair2(mG, gPKShare)
      3. 如果pair1等于pair2，则存储计算出的Reid+1



随机数生成简明图

