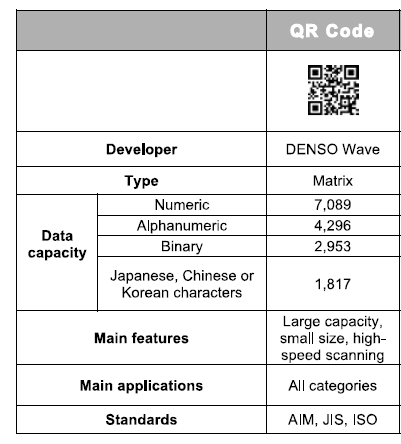
**基于以太坊的动态安全二维码生成方法及系统**

**Keywords：以太坊、动态、安全、二维码**

1. 生成二维码：
2. 在生成二维码的时候，分配一个唯一的“ID”（类似于数据设计中的 Primary Key,保证在全局情况下的唯一性）；
3. 采用ID +Content 的方式，生成一个Hash值,称为Hash1，再用ID+Hash1生成Hash2（Hash2值用于扫码比对安全性）；
4. 单独把ID+Hash1 生成一个二维码，而把二维码Content和Hash2值存放在以太坊的平台中，这样一来，二维码的图片中，存的只有二维码的的ID信息和Hash1，而实际的二维码内容是存在以太坊平台中的。

这样做的思路如下：

 ①、如果二维码图片存放ID+Content的情况下，因为ID是全局唯一的，随着生成次数的增加，ID 的长度势必会越来越长，而二维码所能容纳的长度是有限的，所以能存储的content的内容就会越来越少，所以采取只存储二维码ID+Hash1的方式，假设hash1的长度是固定的n,理论上可以存储 个二维码，因为n 的值比较小，理论上接近无穷大了。

②、在扫码的时候，实际上是扫到的是二维码的唯一ID+hash1，扫码获取到的ID+hash1\_1再用Hash算法，生成这个ID + hash1\_1的Hash值，称为Hash3,再通过ID去以太坊取到之前存储的Hash2,判断Hash3是否等于Hash2，据此判断二维码的安全，如果是安全的，再进行后续操作，如果是不是就丢弃掉。

1. 可以增加的点：  
   1. 、哈希运算方法有好多种，可以都写在论文中，也可以就此分析不同哈希 算法的不同点，针对本文哪种哈希算法是最优的；
   2. 、以太坊平台的介绍也可以写好多了；
   3. 、哈希运算可以放在多线程的算法里面；
   4. 、生成二维码的效率及其安全性检测；
   5. 优缺点分析

优点：1、理论上以太坊可以保证安全

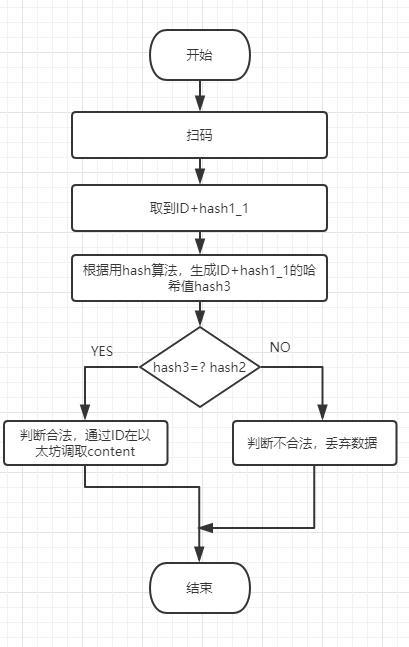
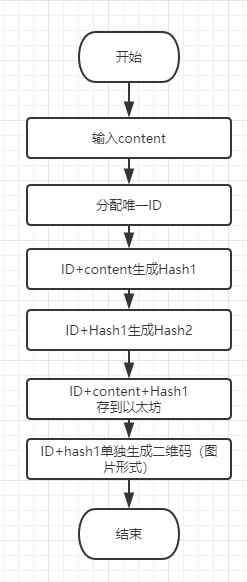
2、由于ID和二维码是分开的，理论上二维码的长度不会受限制，因为二维码不是存在图片上的，都是存在以太坊平台上，（可能以太坊平台会有长度限制，这里不做赘述，以以太坊的为准）

3、采用**两次哈希运算**，可以保证一个ID只能存储一个二维码，只有当ID和Hash1\_1 都匹配的的时候，才能正真调取到实际content

4、二维码的content始终是隐藏的，值存在以太坊中，可以保证不被窃听和篡改。

缺点：1、由于只是把ID生成二维码，在ID长度小的情况下，效率是有提高的，随着长度的增加，效率是在越来越低的。

3、流程图：



生成二维码 扫码

More details:

<https://www.processon.com/view/5b739ce7e4b0f8477db2f751>

<https://www.cnblogs.com/cxyzt/p/8876075.html>