- **利用变色龙哈希算法实现可修改的区块链**

区块链(BlockChain)是一种去中心化的分布式存储系统，传统的区块链都有不可修改的特性，这在很大程度上保证了区块链存储的安全和可信。

但是应用在存储密钥时，这样不可修改的特性就可能导致一些问题，最主要的问题就是无法撤销密钥：当一个密钥被打包成交易存储在链上后，就再也无法修改，永远可以查询到，为了标明密钥的有效性，需要额外的管理成本。

可修改的区块链便可以解决密钥撤销复杂的问题。区块链的不可撤销特性是由梅克尔树(Merkle trees)保证的，即每个区块都要引用前一个区块的哈希值，当某个区块被更改后，它的哈希值就会发生变化，便会导致后续所有区块都不在有效。

为了保证区块链安全可信的同时，又要使它可修改，便需要从区块链的梅克尔树结构入手，能不能使区块做出更改时，哈希值不变呢？一般的哈希算法都是难以寻找碰撞，难以在不改变哈希值的条件下修改内容，但是变色龙哈希可以做到这一点。

变色龙哈希算法(Chameleon Hash)，由 Krawczyk H 与 Rabin T 于 2000 年提出，与传统哈希函数最大的不同的在于，变色龙哈希函数有人为的陷门，当掌握了陷门密钥时，就可以随意找到碰撞，即可以在不改变哈希值的情况下修改内容。

当然，一般的变色龙哈希在陷门密钥泄露后，便失去了意义，区块链作为一种去中心化的系统，没有信任中心来存储陷门密钥，怎么办呢？我选择了把密钥公开，采用逻辑的方式保证区块链的可信和安全性，如下：

变色龙哈希函数的输入值有两个，一个是消息 m，一个是随机序列 s，最后结果 h(m,s)由二者共同确定，需要修改消息内容，即 m->m’ 时，利用陷门密钥，就可以计算出新的随机序列值 s’ ，使得 h(m’,s’)等于 h(m,s)，即获得了碰撞。基于这样的特点，应用在区块链中时，我选择将作为哈希函数输入的交易内容拆分成两部分：一部分是密钥的具体信息，例如公钥，生效时间，有效期等，作为 s 输入；一部分是密钥的有效性信息，只有两种取值(v1 为有效，v0 为无效)，作为 m 输入。这样，交易在需要撤销时，就可以将密钥有效信息置为 v0，生成新的密钥具体信息（当然生成的都是没有实际意义的序列，但由于密钥已经失效，所以无影响）。对于试图在不改变密钥有效性的情况下修改密钥具体信息的恶意攻击者，由于密钥有效位上不为 v1 的取值都是无效的，而修改密钥具体信息必然伴随着密钥有效位的改变，所以他们无法得逞。

- **利用可修改的区块链搭建面向 ICN 的密钥管理系统**

密钥管理系统，尤指在公私钥密钥下，储存用户公钥并提供认证，验证和撤销服务的系统，在物联网 IoT，信息中心网络 ICN 以及其他一些场景中都有广泛的应用。

常见的实现方法有：中心化存储管理，数字证书认证等。中心化的管理显然不适合大型的系统，数字证书认证也有明显的弊端，主要在于每个数字证书系统需要可信的根密钥来构建信任链，作为可信中心的根密钥持有者，面临着维持信任成本，保护密钥泄露等重要责任，一旦根密钥被破解或被泄露，整个系统便会进入不安全状态。

而区块链作为一种去中心化的分布式存储系统，与密钥管理有着良好的相性。就以 ICN 网络为例，每个基站可以作为一个节点，共同维护一个区块链，从而就不需要一个中心化的服务器，用户需要获取服务时，向最近的基站发起请求即可，基站根据用户需求对区块链进行相应存储或搜索即可。这样的系统抗攻击性能也较好，恶意攻击者需要攻击网络中超过 51%的节点，才可以破坏这个区块链系统，相比于攻击单个中心服务器，这显然是比较困难的。而且作为一个服务器-客户端架构的系统，可以通过基站的分布来维护负载均衡，性能较好。

- **项目实现**

在具体实现时，首先，区块链基本结构我选择了前一节所述的可修改区块链，每个需要认证的密钥及其相关信息，作为一个交易打包进区块链，交易中包含一个有效位，用来计算变色龙哈希值，撤销时，将有效位置 0 生成碰撞即可。

其次，在共识算法方面，我选择使用 raft 协议，raft 协议与比特币中使用的 pow 相比，不需要额外的算力开销，也可以保证在部分节点故障的情况下，整个系统依旧可以运转良好。

最后，在网络通信层面，最初设计的是双层网络，一层用于数据通信，一层用于同步控制，可以充分利用不同信道的性能差异，后来为了简洁性，合并为一层网络实现了。目前使用的是 http 协议进行节点间通信和客户端服务器交互，后期可以针对不同的传输层进行其他实现。