1. 系统初始化：输入，生成全局参数和密钥。
2. 系统参数

* 随机选择一个大素数p作为群的阶。
* 选择循环群G1和GT，这两个群的阶都是p。
* 选择两个生成元g和h，两者均为G1的元素。
* 选择配对函数e:G1×G1→GT。（e为双线性映射）

1. 密钥生成中心的密钥对：KGC生成系统的密钥对，包括私钥和公钥

* 选择一个随机数作为密钥生成中心的私钥（是模p的整数集合，表示为 {0,1,2,…,p−1}）
* 计算公钥

1. 搜索服务器的密钥对：搜索服务器生成自己的密钥对，包括私钥和公钥​

* 选择一个随机数作为密钥生成中心的私钥
* 计算公钥

1. 数据所有者和数据用户的密钥对

* 数据所有者选择一个随机数作为私钥，计算公钥
* 数据用户选择一个随机数作为私钥，计算公钥

1. 身份双向认证
2. Do-Identity (CID) 生成

* **计算双线性配对**k：使用双线性配对函数e，计算k： 。其中，是数据所有者的私钥，是数据用户的公钥，。
* **计算哈希值 h：**使用哈希函数H1计算哈希值h：。其中，是数据所有者的公钥，是数据用户的公钥。
* **计算 CID**：使用和h计算CID。其中，是数据所有者的私钥。

1. Du-Identity (TID) 生成

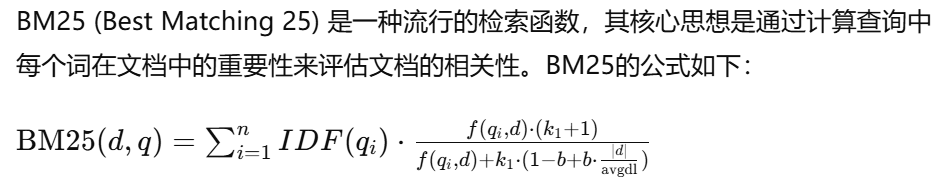
* **计算双线性配对**k：使用双线性配对函数e，计算k： 。其中，是数据所有者的私钥，是数据用户的公钥。
* **计算哈希值h：**使用哈希函数H1计算哈希值：。其中，是数据所有者的公钥，是数据用户的公钥。
* **计算TID**：使用和h计算TID。其中，是数据用户的私钥。

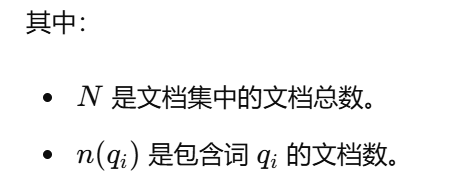
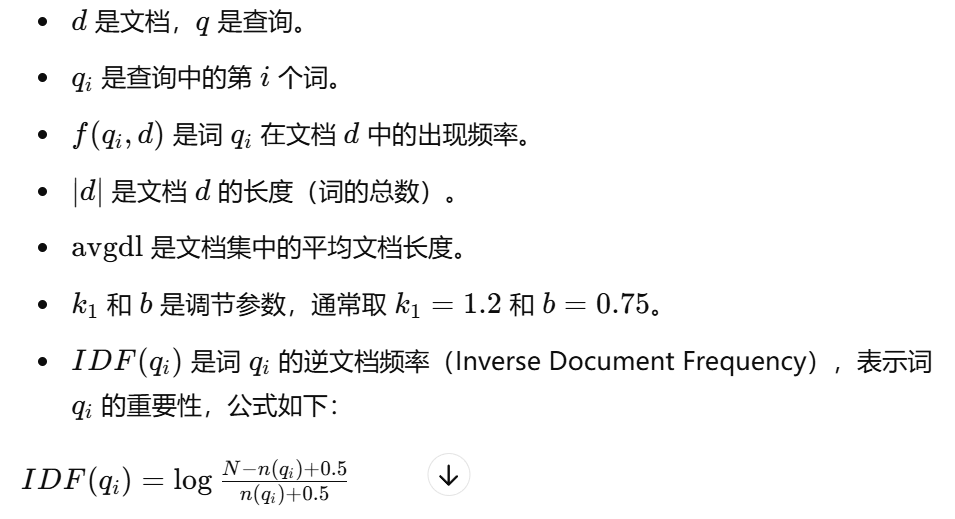
1. 身份认证

计算和，验证两者是否相等，若相等则认证通过，否则认证失败。

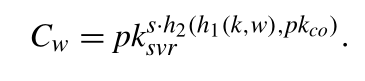
1. 索引构建
2. **创建矩阵索引**：根据关键词和文档的对应关系，创建一个矩阵，其中矩阵的行表示关键词，列表示文档ID，矩阵的值表示关键词在文档中的相关性评分。

如果关键字数量超过400个，则是长文档;否则，它就是一个简短的文档。利用BM25计算关键字wi对应的短文档SDocj的相关分数，利用BM25L计算关键字wi对应的长文档LDocj的相关分数。





1. 加密矩阵索引生成：

* 对矩阵索引中的关键词部分进行加密。（k为CID里的）
* 使用公钥加密算法对矩阵索引中的文档ID部分进行加密。
* 使用VFE-plus编码对矩阵的相关性评分部分进行加密，如将其转化为二进制后三个数字为一组转化为八进制然后把位置倒转。

（示例：两个整数14和24。14的二进制是000 001 110，24的二进制是000 011 000，分别对应八进制的016和030。所以14的VFE是016，24的VFE是030。为了更好地保护数据隐私，我们将VFE.enc，前者的Enc为610，后者为030。）

1. 陷门生成

· **计算关键词哈希值**：使用哈希函数计算待检索关键词的哈希值。1721465841848（k是TID里的）

1. 密文检索
2. 匹配算法：根据用户的检索关键词和数据所有者上传的加密矩阵索引，匹配出相关的密文。**输入**：加密矩阵索引中的关键词部分Cw、用户的检索陷门Tw′、搜索服务器的私钥 sksvr​**过程：**计算Cw是否等于，若相等，则设置搜索向量v的对应位置为1，否则为0。搜索向量v中每个位置表示用户的检索需求中是否包含关键词。
3. 相关性评分和计算：根据搜索向量和加密矩阵索引计算每个密文的总相关性评分。

**输入**：搜索向量v、加密矩阵索引中的数值部分。

**过程：**对于矩阵索引中的每一列，计算搜索向量与该列的内积得到每个文档对应的相关性评分。

1. 阈值比较：比较每个文档的相关性评分和设定的阈值，确定满足条件的密文编号。

**输入：**相关性评分、检索阈值VRrt（由数据用户生成）。

**过程：**如果 TDocj≥VRrt，则输出对应的密文编号 CDocj

1. 云服务器检索：根据相关性评分返回符合条件的密文。

**输入：**满足条件的密文编号CDocj

**过程：**根据密文编号从云服务器获取对应的密文，返回给用户。

6.密文解密