SCD 隐私保护解决方案 Demo 流程

Demo 代码流程：

1. 证书机构发布证书模板以及
2. 用户填写自身相关信息，并附上模板提交给证书机构请求签发证书
3. 证书机构验证用户信息正确，签发认证后的证书
4. 为防止证书机构对认证凭证的使用进行跟踪，用户对认证后的凭证进行混淆，获得混淆凭证
5. 用户选择是否披露相应信息
   - 仅在验证通过后公布披露信息
   - 公布的信息名称会储存在规则集（rule set）的一个数据结构中
6. 生成证明： 输入（混淆证书，用户输入信息，证书模板，规则集，用户私钥）
   - 规则集：事先定义的断言（Predicate）的集合

     断言：Attr_name, type, value

     Type: EQ, GE, GT, LE, LT

   (1) 根据证书规则提取用户输入信息

   (2) 遍历规则集合中所有断言并于用户提供信息逐一比对

   ① 若断言类型为 EQ：

   1) 首先比较用户输入值与断言值是否一致，不一致直接返回错误

   2) 若正确，则判断是否需要标记为揭露类型

   ② 若断言类型为 GE, GT, LE, LT：

   1) 记录待证明断言

   2) 并记录需要披露的断言的 Attr_name

   (3) 生成初始证明：

   ① 验证提取的断言与根据证书提取的用户输入信息的一致性；验证需揭露属性的名称的一致性

   ② 初始化未撤回证明（用于证明证书并未被签发机构撤回）

   ③ 分别为不同的断言类型生成初始证明（init_proof）

   (4) 将 init_proof 与验证者提供 nonce 整合生成最终的 proof
7. 验证者验证生成的证明

   (1) 验证提取的断言与根据证书提取的用户输入信息的一致性；验证需揭露属性的名称的一致性

   (2) 验证未撤回证明（non_revocation_proof）

   (3) 分别验证 eq_proof 和 ne_proof
8. 成功则根据用户所选择的披露信息内容返回相关数据

【注】证明的生成相关详见 AnonCred.pdf

VCL 公开可验证密文账本 Demo 流程

Demo 支持场景：

支持验证 A+B=C; A*B=C; A >= 0

1. 关于范围证明，所使用的都是 Bulletproof
2. Prover 的证明生成都是基于 A 和 B 的承诺以及 C 的随机因子产生
3. Verifier 结合 A、B、C 的密文对 Prover 提供的 proof 进行验证
4. 基于 Curve25519 上的一个素阶群 Ristretto 来实现对于乘法和加法的验证；对应的库是 https://github.com/dalek-cryptography/curve25519-dalek
5. A、B、C 的密文（secret）是以 $A'=g_1^a * g_2^{r_a}$，$B'=g_1^b * g_2^{r_b}$，$C'=g_1^c * g_2^{r_c}$进行存储（其中的 r 定义为 blinding_value，用以隐藏原始值 a）
6. 原始值生成的 credit 则是一个 commitment，对应于椭圆曲线上的一个点