SD04630230: E-Cash and Blockchain **Oct 30, 2023**

实验 1: 哈希函数与零知识证明

Lecturer: 钱宸 (Chen Qian)

1.1 Python 基础

本次实验基于 Python 以及 ecc-pycrypto 库来实现。其中我们将使用 ecc-pycrypto 库中的 P256 椭圆曲线的一个简单实现。

其中 P256 椭圆曲线源代码取自：`https://github.com/lc6chang/ecc-pycrypto`。

使用方法，下载并解压到所在目录。进入 ecc-pycrypto 文件夹，并运行 “pip3 install .”。

仔细阅读 bn256.py 的源代码，理解每个函数之间的关系。熟悉椭圆曲线上的运算操作。

几点提示：

- 使用”from ecc.curve import P256, Point” 加载 P256 曲线
- 椭圆曲线的生成元定义为 “P256.G”
- 椭圆曲线的阶的大小为”P256.n”
- 随机产生一个阶数为 “x = random.randint(1,P256.n)”，需要在最开始 “import random”
- 椭圆曲线上每个点的字符表示可以写成 “repr(point)”

1.2 哈希函数

在课上我们知道了抗碰撞的哈希函数可以由离散对数假设得到。

问题 1: 利用 bn256 库，构造抗碰撞的哈希函数。

提示: 包括 $\mathsf{Setup}() \to \mathsf{K}$ 和 $\mathsf{Hash}(\mathsf{K}, \mathsf{m}) \to \mathsf{h}$ 两个函数。

我们证明了如果离散对数问题是困难的，那么找到问题 1 中的哈希函数的碰撞是一个困难问题。那么在知道哈希函数密钥 $\mathsf{K} = (g, h)$ 中离散对数 $k = \log_g(h)$ 的情况下，能够产生哈希函数的碰撞呢?

问题 2: 构造两个新的函数 SetupInsecure 和 FindCol。其中 SetupInsecure 不仅产生哈希函数的密钥 K，也输出离散对数 $k = \log_g(h)$。FincCol 函数利用离散对数找到一组哈希函数的碰撞。

- $\mathsf{SetupInsecure}() \to (\mathsf{K}, k)$
- $\mathsf{FindCol}(\mathsf{K}, k) \to (x_1, x_2)$

测试：找出的 (x_1, x_2) 是否满足哈希函数碰撞的定义。

1.3 零知识证明

在课堂上，我们给出了离散对数的非交互式零知识证明。

问题 3：实现离散对数的非交互式零知识证明。

- $\mathsf{Prove}(\mathsf{x}, \mathsf{w}) \to \pi$
- $\mathsf{Verif}(\mathsf{x}, \pi) \to \{0, 1\}$

提示：使用 Python 自带的 hash 函数 $\mathsf{hash}(\mathsf{Any}) \to \mathsf{int}$

（DDH 的零知识证明）我们在之前的课程中学过判定性 Diffie-Hellman 假设，即给定 $(f, g, F, G) \in \mathbb{G}^4$ 四个不同的群元素，判断是否存在 $r \in \mathbb{Z}_p$，使得 $(F, G) = (f^r, g^r)$。

问题 4：令 $(f, g, F = f^r, G = g^r) \in \mathbb{G}^4$，写出 (f, g, F, G) 为 DDH 对的**交互式** 零知识证明（此处不需要编程），并证明在离散对数假设下，如果 (f, g, F, G) 不存在 DDH 关系，则能够给出可通过验证的证明等价于能解决离散对数难题。

提示：如果 $(F, G) = (f^r, g^r)$，则我们知道 $F/G = (f/g)^r$。

问题 5：编程实现问题 4 中零知识证明的非交互式版本。