

步骤:

1. 实现抽象类 `Authentication` 的成员变量和函数:

(1) 3 个必须的 参数: `self.attackDataSet` (用于攻击的数据库)、`self.victim` (受害者)、`self.limitedTryNum` (最多尝试攻击次数)。根据需求有额外的参数可以自行添加。

(2) 1 个函数: `oneSampleTest(self, attackSample)`, 用一个 `attackSample` 对对应模型攻击, 如果通过返回 `True`, 否则返回 `False`。

2. 确定受害者和用于攻击的数据库 (包含除受害者以外所有用户的 `cycle`, 每个 `cycle` 为 $3*N$ 的矩阵, 样例在 `unitTestAdversary.py` 里)

3. 实例化 `Authentication` 对应的类

4. 实例化 `Adversary` 类, 并将 3 中的实例作为参数传入

5. 调用 `Adversary` 实例的 `run` 函数, 返回 `tryNum` 和 `pos`。如果在有限次数内攻破, `tryNum` 表示第几次攻破, `pos` 表示攻破的数据为 `attackDataSet` 的第几个数据; 如果没有通过, 两个值都为-1.

PS: 具体样例可以参考 `unitTestAdversary.py`