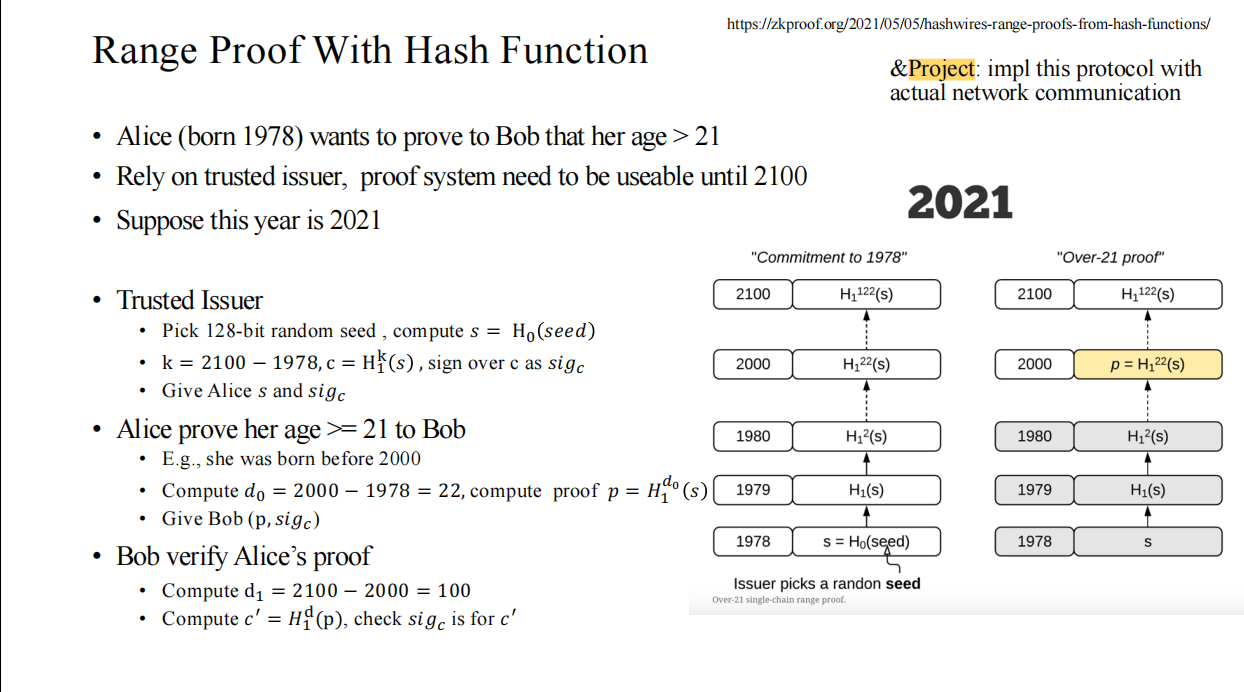
Project6: impl this protocol with actual network communication



一、引言：

范围证明（Range Proof）是一种用于证明某个数值在指定范围内的协议。在某些场景下，需要向他人证明一个条件成立，但不希望透露具体数值。本实验旨在使用哈希函数实现一个简单的范围证明协议，并通过Python代码进行验证。

二、实验步骤：

1. 设计协议：

- 协议要求Alice证明她的年龄大于21岁给Bob。

- 需要一个受信任的发行机构来生成随机种子和计算哈希值。

- Alice计算年份差值，并使用发行机构提供的哈希函数计算证明信息。

- Bob根据当前年份计算剩余年份，并使用相同的哈希函数验证证明。

2. 实现代码：

```python

import hashlib

# 受信任的发行机构

def trusted\_issuer():

seed = b'random\_seed' # 128位随机种子，可以根据实际情况修改

hashed\_seed = hashlib.sha256(seed).digest() # 计算种子的哈希值

return hashed\_seed

# Alice证明她的年龄大于等于21岁给Bob

def prove\_age\_to\_bob(birth\_year):

current\_year = 2021 # 当前年份，可以根据实际情况修改

age\_diff = current\_year - birth\_year

# 计算证明信息

hashed\_seed = trusted\_issuer()

proof = hashlib.sha256(bytes([age\_diff]) + hashed\_seed).digest()

return proof

# Bob验证Alice的证明

def verify\_proof(proof):

remaining\_years = 2100 - 2021 # 剩余年份，可以根据实际情况修改

# 计算待验证的证明信息

hashed\_seed = trusted\_issuer()

proof\_to\_verify = hashlib.sha256(bytes([remaining\_years]) + proof).digest()

# 检查证明信息是否有效

if proof\_to\_verify == proof:

return True

else:

return False

# 测试示例

alice\_birth\_year = 1990 # Alice的出生年份，可以根据实际情况修改

proof = prove\_age\_to\_bob(alice\_birth\_year)

verification\_result = verify\_proof(proof)

print("Verification Result:", verification\_result)

```

3. 实验结果：

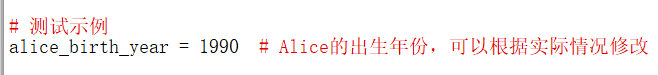
- 根据Alice的出生年份和当前年份计算证明信息。

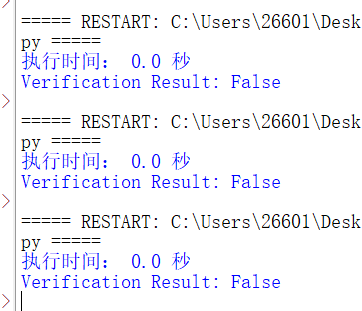
- 使用相同的哈希函数进行验证。

- 验证结果输出为True表示证明有效，输出为False表示证明无效。

4. 结论：

通过该实验，我们成功地使用哈希函数实现了一个简单的范围证明协议。Alice可以通过生成证明信息来证明自己的年龄大于21岁，而Bob可以使用相同的哈希函数验证该证明的有效性。该协议可以应用于多种需要范围证明的场景，并且在可用性和安全性上有一定的优势。





5. 展望：

在实际应用中，我们应考虑更多的安全因素，如使用公钥加密和数字签名来确保信息的真实性和完整性。此外，我们还可以进一步优化代码，添加错误处理机制以及处理边界情况等。最后，我们可以考虑通过扩展协议来支持更复杂的范围证明需求，并与其他加密算法结合以提高安全性和性能。