区块链隐私保护和数据安全

唐韶华

华南理工大学

2018年9月

CONTENT

- 区块链本质上是一个类BFT的系统，需要不同节点对交易以及状态进行验证重算来达成共识，因此要求链上数据都是非加密且公开的，造成了数据的隐私问题。
- 区块链中的数据提供方并不希望数据完全公开，尤其是敏感数据，包括交易身份、交易金额、合约等。

隐私场景：

1. 用户利用区块链上的数字资产向商户购买商品，但并不希望向所有节点暴露自己的消费历史
2. 两家公司在区块链上签订了股权相关的智能合约，但并不希望马上披露以免引发股价异常波动
3. 销售商并不希望在区块链上披露自己的供应商，否则可能造成竞争加剧以及自己的成本上升

- **区块链的隐私问题：**
- 1 .假名与匿名

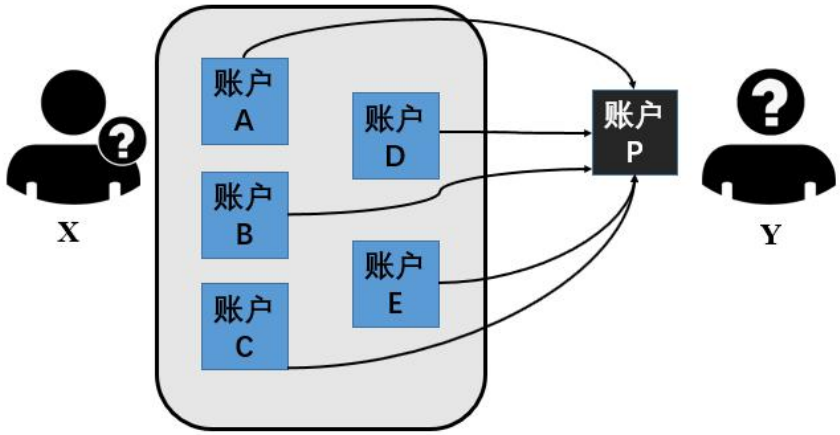
2. 合约代码隐私

3. 合约输入隐私

- 1．“假名“与”匿名“

- 假名：比特币，以太坊等系统中，不使用真名，而是用钱包地址代表使用者身份。

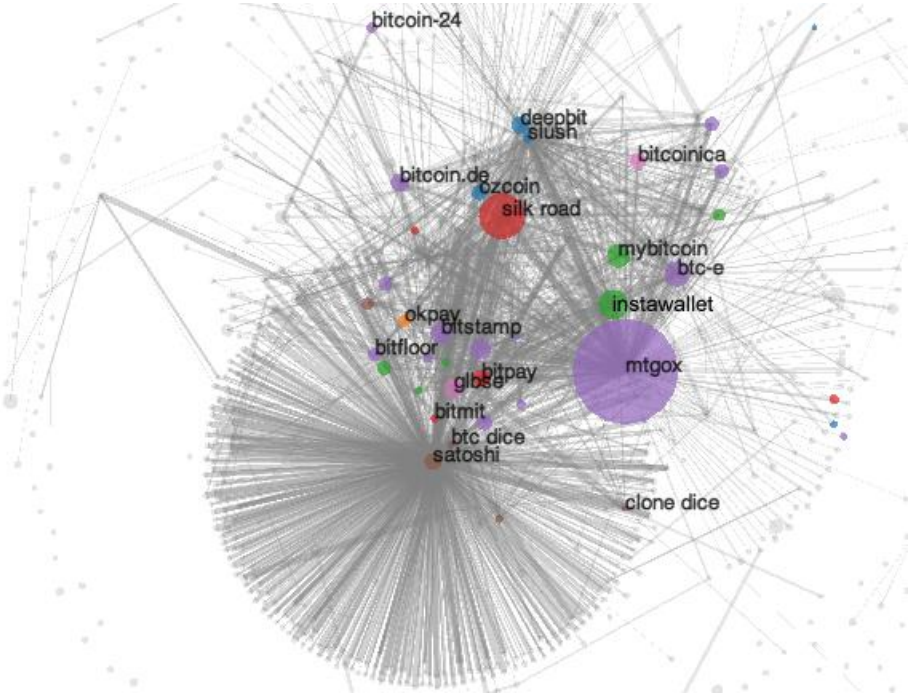
- 匿名：具备无关联性的假名。比特币等容易受到去匿名攻击，因此不具备匿名性。



同一用户的多个地址关联到同一地址簇

- 去匿名攻击，利用了统计规律和背景知识攻击。

- 1. 根据大量交易记录，对交易地址进行关联，得到大量地址簇。
  2. 根据其中已知信息的地址，比如交易所，某些虚拟货币商家等，对其进行标签化。
  3. 再结合现实中的背景知识，给个人用户的地址簇打上特征标签，得到交易分析图来进行去匿名化，包括比特币财产数额，交易偏好等。



2. 合约代码隐私

由于以太坊合约的编译后生成的EVM字节码是全网公开，因此可以通过反编译的方法获得智能合约的Solidity语法代码。

```
[     0x0] | PUSH1              | ['0x80']
[     0x2] | PUSH1              | ['0x40']
[     0x4] | MSTORE             | None
[     0x5] | PUSH1              | ['0x4']
[     0x7] | CALLDATASIZE       | None
[     0x8] | LT                 | None
[     0x9] | PUSH2              | ['0x61']
[     0xc] | JUMPI              | None
[     0xd] | PUSH4              | ['0xffffffff']
[    0x12] | PUSH29             | ['0x100000000000000000000000000000000000000000000000000000000
0000000000']
[    0x30] | PUSH1              | ['0x0']
[    0x32] | CALLDATALOAD       | None
[    0x33] | DIV                | None
[    0x34] | AND                | None
[    0x35] | PUSH4              | ['0x6b59084d']
[    0x3a] | DUP2               | None
[    0x3b] | EQ                 | None
[    0x3c] | PUSH2              | ['0x66']
[    0x3f] | JUMPI              | None
[    0x40] | DUP1               | None
[    0x41] | PUSH4              | ['0x8da5cb5b']
[    0x46] | EQ                 | None
[    0x47] | PUSH2              | ['0xa4']
[    0x4a] | JUMPI              | None
[    0x4b] | DUP1               | None
[    0x4c] | PUSH4              | ['0xcaf44683']
[    0x51] | EQ                 | None
[    0x52] | PUSH2              | ['0xb9']
[    0x55] | JUMPI              | None
[    0x56] | DUP1               | None
[    0x57] | PUSH4              | ['0xe3ac5d26']
[    0x5c] | EQ                 | None
[    0x5d] | PUSH2              | ['0xd3']
[    0x60] | JUMPI              | None
[    0x61] | JUMPDEST           | None
[    0x62] | PUSH1              | ['0x0']
[    0x64] | DUP1               | None
[    0x65] | REVERT             | None
```

```
def main():
    if CALLDATASIZE >= 4:
        data = CALLDATA[:4]
        if data == 0x6b59084d:
            test1()
        elif data == 0x8da5cb5b:
            owner()
        elif data == 0xcaf44683:
            test2()
        elif data == 0xe3ac5d26:
            prize()
        else:
            pass
    raise
```

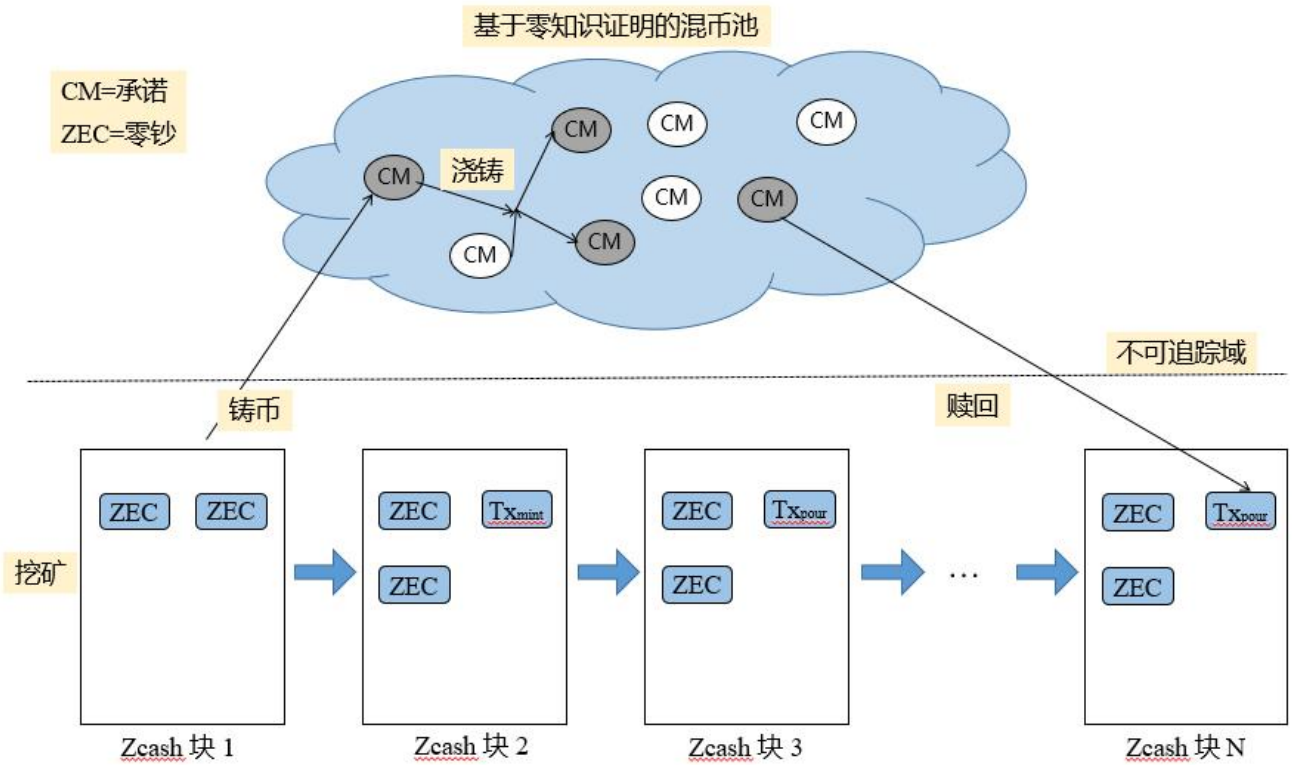
- 3. 合约输入隐私

- 对于以太坊来说，当一个用户调用合约，并且成功广播到全网的时候，其调用参数是明文存储的，这对于用户的数据安全也是一种泄露。

  举例，在一篇利用区块链进行数据存储的论文中，作者利用智能合约存储了用户密码的Hash值，再利用智能合约计算Hash值进行用户认证。乍看之下，合约中只存储了Hash值，但是调用合约的参数实际是密码明文，可以从交易的Data字段直接读取出来。

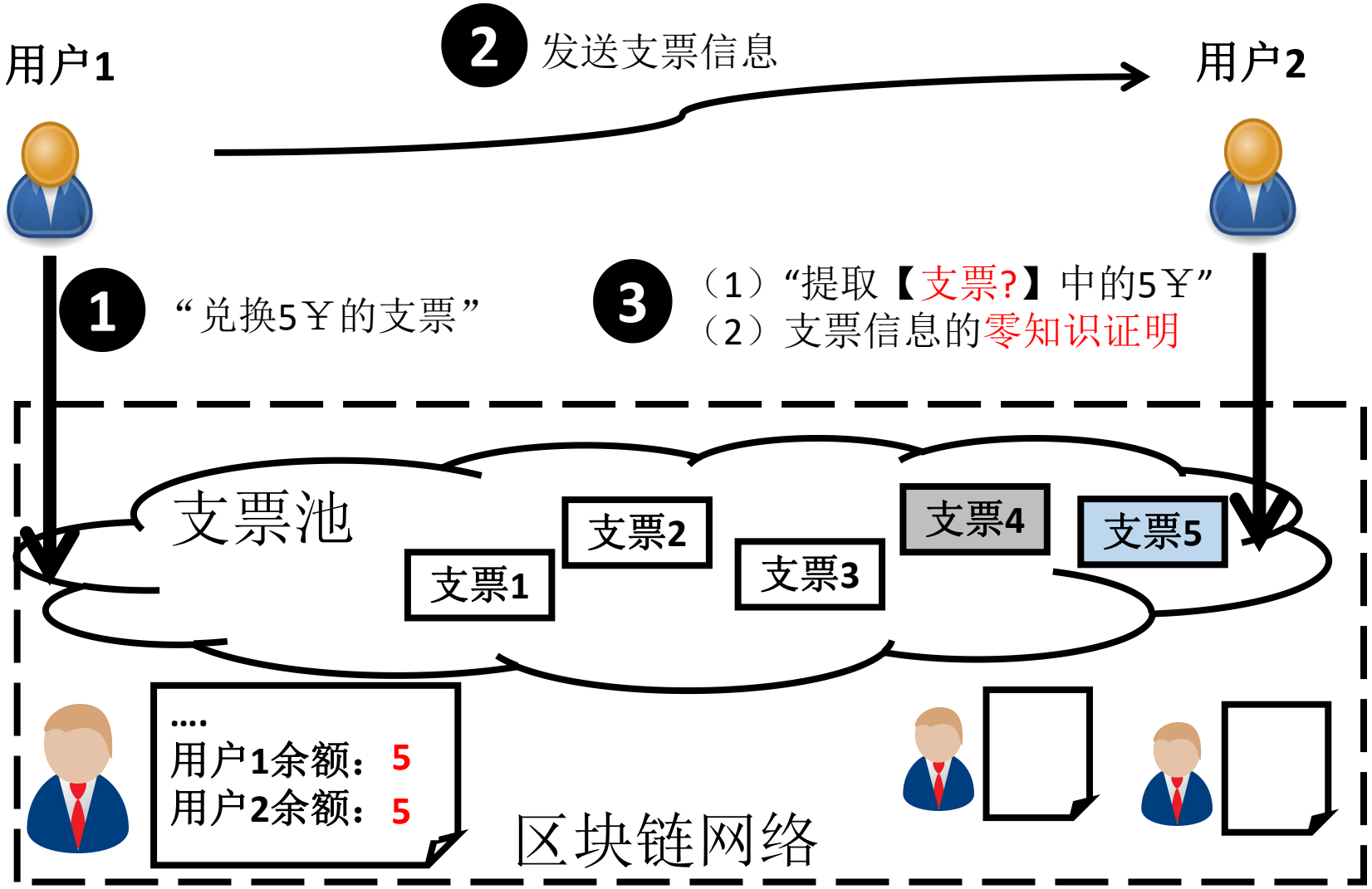
基于zkSNARK的混币池

- 1. 零钞（Zcash）是一种利用零知识证明来构造一种完美混币池的密码学货币。
- 零知识证明与zkSNARK



基于zkSNARK的混币池

2. 零钞运行的原理



Sasson, Eli Ben, et al. “Zerocash: Decentralized anonymous payments from bitcoin.” **S&P 2014**

保护合约数据私密性

- 零钞方案针对交易双方的身份，交易金额进行隐私保护，不能保护合约的输入数据。
- Hawk：采用了与Zcash类似的零知识证明模式来运行智能合约，从而实现了匿名性
- 利用**安全多方计算**来保证合约执行过程中的数据隐私尤其是输入隐私。

Kosba, A., Miller, A., Shi, E., Wen, Z., & Papamanthou, C. Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts. *Proceedings - 2016 IEEE Symposium on Security and Privacy*

保护合约数据私密性

- Hawk 项目基于以太坊的智能合约平台进行开发，合约语言是Serpent。
- 论文中的示例代码分为**公有合约部分**与**私有合约部分。**
- **私有合约：**
  - 负责处理用户的输入
  - 结合多方计算以及零知识证明来隐藏用户身份以及具体的输入值

公有合约：

- 提供押金的逻辑，保证交易方不能在中途退出

```
1  HawkDeclareParties(Seller,/* N parties */);
2  HawkDeclareTimeouts(/* hardcoded timeouts */);

3    // Private portion φpriv
4  private contract auction(Inp &in, Outp &out) {
5    int winner = -1;
6    int bestprice = -1;
7    int secondprice = -1;

8    for (int i = 0; i < N; i++) {
9      if (in.party[i].$val > bestprice) {
10       secondprice = bestprice;
11       bestprice = in.party[i].$val;
12       winner = i;
13     } else if (in.party[i].$val > secondprice) {
14       secondprice = in.party[i].$val;
15     }
16   }

17   // Winner pays secondprice to seller
18   // Everyone else is refunded
19   out.Seller.$val = secondprice;
20   out.party[winner].$val = bestprice-secondprice;
21   out.winner = winner;
22   for (int i = 0; i < N; i++) {
23     if (i != winner)
24       out.party[i].$val = in.party[i].$val;
25   }
26 }

27   // Public portion φpub
28 public contract deposit {
29   // Manager deposited $N earlier
30   def check():  // invoked on contract completion
31     send $N to Manager  // refund manager
32   def managerTimeOut():
33     for (i in range($N)):
34       send $1 to party[i]
35 }
```

Kosba, A., Miller, A., Shi, E., Wen, Z., & Papamanthou, C. Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts. *Proceedings - 2016 IEEE Symposium on Security and Privacy*

保护合约数据私密性

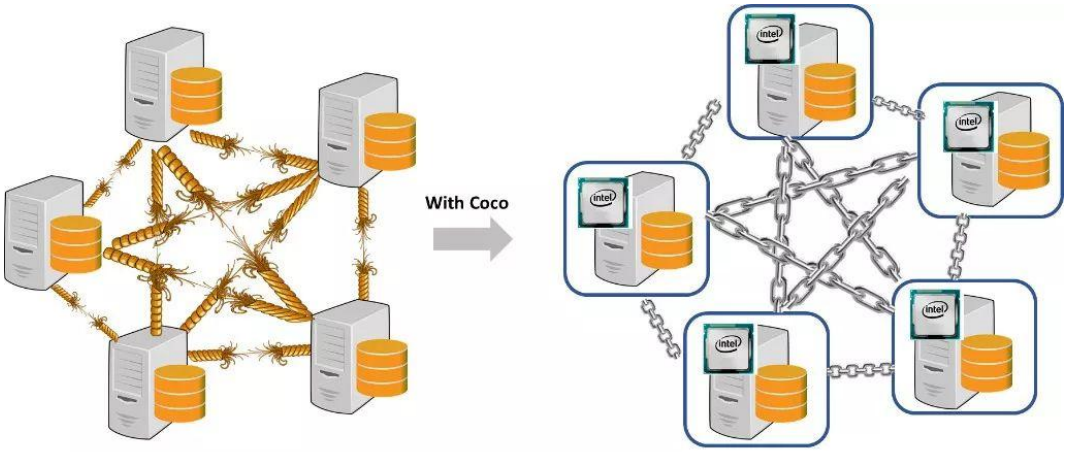
- Hawk 局限性：
  - 多方计算的效率比较低下
  - Hawk 仍旧不能保证合约代码的私密性，仍可以通过反编译等手段获得合约代码。

兼容任意区块链

- Coco 理论上可以用来保护任意区块链系统的隐私性

- Coco Framework不是独立的块链协议，它提供了一个信任的基础，可以整合现有的块链协议(如Ethereum等)提供完整的企业级区块链解决方案。

Coco Framework充分利用**可信计算环境**TEE (Trusted Execution Environment)，如 Intel SGX 和Windows 虚拟安全模式(VSM)，创建了可信的网络。



兼容任意区块链

可信计算环境 TEE

可以证明放入代码的正确性 → 能保证运行时内部数据对外界不可见以及不被篡改 → 可以确保保障区块链协议关键代码和数据的机密性、完整性 → 使得区块链的应用可以在完全受信任的成员节点上高效运行

Coco Framework运行的原理

1. Coco Framework搭建的网络中的节点，通过证书的验证(如Intel背书)而成为可信节点VN(Trusted Validating Nodes)。
2. 每个节点运行Coco Framework和某个区块链的协议(比如以太坊)，并根据所选取的一致性协议系统选取lead来处理应用中的交易事务。
3. 成为lead的VN就像以太坊里面的矿工，但不同的是Coco Framework里面的每个VN都可以通过TEE attestation 验证其他节点执行时候所用的代码哈希值(恶意行为将直接被发现)，而不需要像以太坊一样通过重新计算交易来验证。
4. VN之间通过TEE可以互相验证身份和代码从而建立可信的连接。
5. Coco Framework包含了一套密钥及权限管理机制，可保证只有在TEE中才能处理加密后的交易，并且只有拥有相应权限的用户才能查看相关状态。

兼容任意区块链

- Coco Framework优点
- 简而言之就是解决了**性能**，**隐私**以及**组织管理**三大“顽疾”

1. 吞吐量和交易响应时间接近数据库的速度
2. 支持更丰富、灵活的隐私保护模型
3. 提供可编程的管理模型来支持任意的分布式管理策略
4. 支持**非确定性**的交易与运算

兼容任意区块链

- Coco 局限性：

1. 理论上兼容任意区块链，但目前对公有链影响较低
2. 需要特殊的可信硬件，在普及上存在问题

以太坊隐私保护技术

- Coco 主要侧重于对联盟链的优化，并且需要特殊硬件，因此对以太坊等架构影响有限。
- 以太坊团队与零钞团队合作，在以太坊最新版本实现了匿名功能，该版本被称为Baby ZoE（初级版Zcash）。
- 与零钞的区别：
  1. 进行了大量简化，只保留了匿名转账中验证相关的椭圆曲线操作和复用了Zcash中的公共初始化参数；
  2. 除了C++版本的以太坊，其他语言版本的以太坊需要调用特殊的合约来辅助实现匿名转账。
- 局限：
  1. Baby ZoE 降低了匿名转账的复杂性，但对于以太坊来说，还是增加了转账的Gas消耗；
  2. 目前无法构建通用的匿名合约。

小结

隐私方案对比：

名称	适用类型	技术特点	优点	缺点
零钞	公有链	zkSNARK	保护身份数额隐私	参数初始化复杂
Hawk	公有链/联盟链	zkSNARK、多方计算、可信计算	保证合约输入隐私	不保护合约代码
Quorum	联盟链	PrivateFor设定隐私策略、Raft	灵活的隐私策略	引入监管节点
Coco	联盟链	可信计算、Raft	保证合约代码隐私	依赖硬件
以太坊	公有链	Baby Zoe	同零钞	成本昂贵

小结

研究趋势1：去匿名攻击的多样化

- 交易图分析，网络层分析，侧信道分析等

研究趋势2：基于实际场景的隐私策略

- 数据对监督仲裁节点可见，对普通节点是加密状态，例 Quorum,Hyperledger, ChinaLedger等

研究趋势3：采用优化的密码学算法来保护隐私

- 零知识证明、环签名、同态加密、安全多方计算等

小结

- 区块链能简化多个互不信任实体之间的业务流程，但会带来隐私问题
- 利用传统密码学方法能部分解决区块链隐私问题，但目前效率有待提高
- 可信硬件提供了一种解决区块链隐私的新思路，能在安全、隐私、效率上获得一个良好的折衷
- 区块链的隐私问题还有很大的研究空间，需要密码学、网络安全与系统安全等领域的合作

Q & A