

## 隐私保护中的密码学工具

### 欧链科技—区块链高端技术沙龙(二)



扫一扫关注欧链小秘书

添加时请备注“技术社区”

11月25日，欧链科技区块链高端技术沙龙有幸请到中科院信工所副研究员，陈宇博士做《隐私保护中的密码学工具》主题讲座。陈宇博士从密码学家的角度分析了隐私问题，并详细介绍了安全多方计算、同态加密、零知识证明等多种密码工具及其在区块链中的应用。该沙龙采取闭门形式，将陆续邀请国内外区块链知名技术专家做主题报告，并诚邀知名区块链行业专家和企业负责人参与讨论，会后将发布高质量的技术报告。欧链科技立志打造区块链第一技术品牌！

欧链科技：非常感谢大家在周末这样一个大好的时光，来参加我们“区块链高端技术沙龙”第二期的活动，刚刚过去的这三个月，应该算是国内区块链行业一个小小的寒冬，好在我们还是挺过来了。

这一次我们邀请到中国科学院信息工程研究所信息安全国家重点实验室的陈宇博士，给大家介绍一下隐私保护学中的密码学工具。我们上一期里讲到区块链的安全与问题，当时大家对隐私保护特别感兴趣，其实我们可以看到在最近报道、访谈里都会提隐私保护这样的事，我可以保证大家今天下午在这里听到的一定是最听过的隐私保护里干货最多的一次。陈宇博士一直在做密码学理论性的研究，对于密码学的操作工具、应用会一一在汇报里给大家介绍一下。现在把会场交给陈博士，请他给大家介绍一下隐私保护中的密码学工具。

陈宇博士：首先感谢邀请和大家在周末还过来，我一直都是做比较偏理论的研究，

对应用其实知道的很少，我今天讲的所有的应用就是尽我所能讲一些密码学工具与现实贴的比较紧的应用，如果有什么讲的不对的地方，请大家随时指出，我的报告希望大家互动多一些，大家有什么问题随时提，不要等到最后。

我首先简单介绍一下在我理解当中密码学的研究分为这样几个层次，最下面这层是基本的数学理论和复杂性理论，比如 P 复杂性类，和 NP 复杂性类是不是等价。再上面一层各种基本的组件和密码工具，这一层它没有办法直接应用到世纪中，比如代码混淆各类编码等等。在这之上我们会对这些基本的组件做进一步的分装，这一步分装之后得到的密码方案可以直接用到现实世界中，比如对称加密方案、数字鲜明、密钥传输协议、零知识。再上面一层可能是对这一类核心密码解释再进行分装，直接与应用层挂钩，比如 SSL、IPSec、SSH、Kerberos 等等。我自己的研究偏向第二、第三层，我们就开始今天的主要内容。

主要内容分为以下四大部分，

第一部分讲讲数据的隐私，第二部分可能大家之前听到的会相对少一些，就是所谓的密钥隐私。第三部分是计算中的隐私，第四部分是证明中的隐私。

- ➊ Privacy of Data
- ➋ Privacy of Key
- ➌ Privacy in Computing
  - Outsourced Computation
  - Multiparty Computation
- ➍ Privacy in Proof

©Yu Chen, CAS  
2 / 67

21 世纪什么最贵？当时说是人才，其实我觉得话没说完，现在大家都知道，数据为王，其实数据最重要。在现实社会中，小至银行卡的数据、个人医疗信息，上至各种高级的军事机密，他们都是无价的数据。我们怎么保护他们？或者说我们保护数据有哪两个最重要的地方？第一保证数据在存储中是静态安全，第二数据很多时候流动起来这样更要保证它在传输过程中的安全。

怎么保护数据？其实保护数据隐私，最简单的方法，我有一个宝贝，古代怎么做？锁到一个箱子里，其实这样一个过程，就是我在对这个数据施加一个访问控制，只有拥有相应钥匙的人才能解开。数字时代就是通过加密技术，你的数据做访问控制。

我们在说使用什么样的加密方案做访问控制之前，首先要问自己一个问题，为了有的放矢，我们希望达到什么样级别的数据隐私？这个是最重要的。有的时候我们经常会讨论我需要安全，到底要什么？其实我也不知道我到底要什么样的安全。我们最直观的一个想法，我把数据加密之后，我期望的安全性就是，如果没有这个密钥的话，你是没有办法精确恢复出这个加密消息的全部内容，但你想想这种级别的安全性，我们可以把它称为单向性，这是一种非常非常弱的安全性。

举个例子，实际上因为现实当中，这个数据的来源非常多多样性，他每一个比特都可能是非常关键的。比如说我在对这一个消息的第一位、第二位、第 N 位做加密传输的时候，这每一位都可能表示一个，一个标志可能是表示股票是卖还是买，我转账这个第一位是 1 还是 0，这个差距非常大，所以数字世界里每一个比特都非常重要。这个是我想向大家传递的第一个观点，这个也就是密码学里面特别是加密里面一个很重要的安全概念，我期望达到最基本的消息是不泄露一个比特的信息。这个看上去很简单，实际上它的定义也花了很长的时间，1984 年前后，估计他们也想了好多年才想了这个定义。

这个概念这样简单讲一下大家不太明白，我们在理论密码学中怎么来做？通过一个游戏来刻划这个安全性，假设这是一个提手，这是一个挑战者，现在这个提手允许你任意选两个消息，发送，它发送一个币，加密发给 C。就问这个 C，它没有办法以超过  $1/2$  的概率准确猜测出这个  $\beta$  是什么。最好的结果就是一个乱

猜，不知道大家有没有体会到定义巧

Basic Requirement - IND Security

妙的地方，为什么允许提手任意 **M0**、

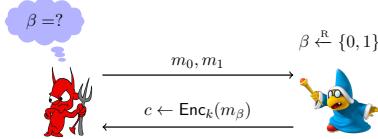
$c$  does not reveal any bit of  $m$

**M1**，任意接近只有一个比特不同，即

使这种情况下都没有办法分辨出来

到底是对哪个消息的加密，所以这就

达到了不泄露任何一个比特的消息。



©Yu Chen, CAS  
7 / 67

之后密码方案，其实加密技术一直演化到现在，整个的演化方向还是相对明确的，概括为就是更高更快更强。我稍微解释一下，这个更高是什么意思，更高指的是密码方案的表达能力更强、功能性更强。更快，先要算得快，第二还要功耗。比如我要在 IOT 上，在一个很小的服务器上跑起来相应的密码程序。第三个就是更强，更强是它能对很多不可预知的攻击都百毒不侵。

我们延着更高的方向梳理一下密码学，特别是加密的发展过程。它主要围绕着访问控制的表达能力上，走一个粗粒度的访问控制变成一个细粒度的访问控制，我前面说加密技术保护数字隐私的最基本的方法就是对这个数据施加一个访问控制，这个访问控制我们其实越细粒度越好，后面会具体讲一下。

这个大家可能会关注比较少，很难意识到。实际上很多时候，现在需要对加密的数据还要做处理。很多时候我在不知道密钥的情况下对加密的数据进行操作，大家如果有相应的私钥的话这个事情没有难度，但这个问题真正难的地方，我需要你在不知道私钥的情况下对数据进行。

我们从最简单的对称密码讲起，加密的双方有一把共同的密钥，用密码加密，接收方只有同样的密钥才能解开。它的好处是简单好用，并且速度快。它的缺点是功能性太差了，比如说如果说想做一个交互的话，保密通信的话，一定需要一

个事先就有的秘密，要么两个人一开始凑的很近，交换一个小秘密，要么之后有一条量子信道，先协调然后再通信。这个在过去可能还可以，现在就不行了，因为现在大家都不知道对方在什么地方。

另外它有一个非常复杂的密钥管理，比如在座的人我们需要两两之间通信的话，那你想想一共之间会产生  $O(N^2)$  量级的密钥。但它在哪里比较有用？我自己加密自己解密，比如所有操作系统上的磁盘加密。所以一定要把需求先列出来，选择适当的工具。

针对这两个问题，我们开始问一个非常大胆的想法，如果这个通讯，它一定需要拥抱同一把 **key** 才能做一个安全的保密通讯或者安全的加密吗？这个结果是大家熟知的，1976 年一篇文章，他们提出了公钥加密这个概念。

公钥加密里面每个用户自己产生一对密钥，**pK** 和 **sK**，如果 **Bob** 想发一条消息的话，直接使用公钥对消息进行加密，公钥就是它张贴在任何地方，谁都可以用得到，它的功能就是做加密。只有 **Alice** 拥有相应的私钥对密码进行解释才能得到相应结果。

这个公钥加密的好处是，没有事先存储一个数据，任何两个用户之间，进行保密通信的时候不需要事先有一个共享的秘密，最核心的一个贡献是什么？自此之后安全的保密性可以在公开的信道上进行，这是非常了不起的结果，就是我不怕你有任何的窃听，没关系，可以在公开的信道上完成。

大部分情况下只要使用公钥加密，都不需要有在线支持，所有的公钥在 **PKE** 里面，就是这样一个数字证书。我看到这个数字证书之后我不知道它跟哪个现实中的实体是有关联的，这就是用这个公链它是不可连接的。

怎么去认证一个所谓的公开钥匙，我们必须借助 **KPI**，在验证这个 **PK** 和 **Sk**

的合法性之后，这个某某人他的公钥确实是这一串数字，我给你盖个章，这个盖章的过程就是做一个数字签名，相当于为一个绑定关系背书。每一次使用这个公钥的时候，比如我在签名的时候，或者是加密的时候，他都要出示这个证书。

举个最简单的例子，比如我们在访问你的 **Gmail** 邮箱的时候，怎么知道我访问的一定是 **gmail**，所以一开始谷歌给我一个数字证书，告诉我是谷歌，同时我的公钥是什么，并且有一个证明，之后才可以用谷歌的公钥进行加密。

这种公钥和实体的不可连接性，恰恰让我们因祸得福。你想像一下这个不受欢迎的特点，但它在比特币里特别受欢迎。就是我没办法看到这个公钥，我不知道谁拥有这个地址。这反而是比特币里面非常希望的一个性质，所以为什么说比特币能想到，就说它能把理论学中诟病的东西用的比较神奇，这是化腐朽为神奇的一个地方。

我要强调一点，这种身份和公钥的不可连接性和匿名性不是一回事，我后面强调一下，这些术语不能通用，只是说我看到它们没办法链接上，并不是说就完全匿名。

#### A Blessing in Disguise

Unlinkable property is desirable in Block Chain  
• unable to trace who owns an address



Unlinkable ≠ Anonymous

**PKE** 一个比较麻烦的地方是它在使用前必须认证，它很强依赖 **PPI** 这种基础设施的部署，同时只要用 **PKC**，它都必须麻烦。所以说传统的这种公钥，一定要有一个证书跟着它。

©Yu Chen, CAS  
15 / 67

它的功能性其实也不是特别强，你要注意到这个私钥没有进一步向下的能力。那有没有一种新型的 **PKE**，它的公钥是自解释并且是私钥可以代理的。

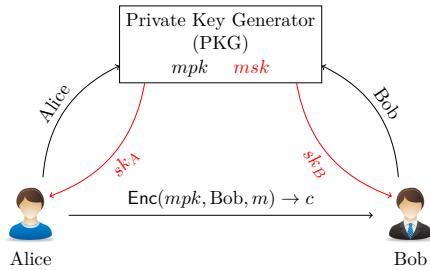
基于身份加密 (**IBE**) 它的一个好处是任何东西，你的邮件、手机号码、地址

都可以作为你的公钥，私钥是你拿着你的信息到一个地方注册一下，会给你一个相应的私钥。

这里有一个好处 PKE 不需要实时在线，还有一个好处，实际上这个可以看成 MSK 向下的代理，相当于主私钥为每一个用户分配了一个私钥，每一个用户其实是从 MSK 抽取出来的。

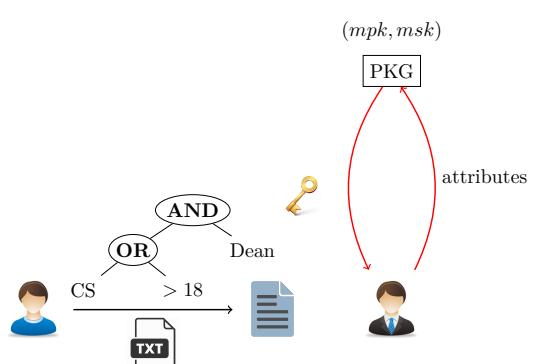
尽管 IBE 比 PKE 进步了一大步，但还有一个缺憾，它的访问控制是一对一的。解释一下什么是一对一的访问控制，这个是在比如身份 ID 下加密的，谁能解密呢？SKid 能加密，加密给我的只有我能来解。有的时候我希望我加密的东西可以主动来解，甚至是符合策略的人来解，这就是在 2005 年前后开始的，这里面把这个泛化成访问控制结构，我加密前先想一下允许哪些人访问，把访问策略写下来，加密。适合属性相关的，当且仅当符合这个访问控制数的时候，才能解密成功。

举个例子看一下比较直观。假设这个 Alice 要加密一个文本文档，他想对这个加密后的消息做一个什么样的访问控制呢？你要么是这个系主任，必须是系主任，同时你要么是 CS，要么是年龄大于 18，才有对这个加密后的消息做解密的能力。这个地方大家



- everything could be used as identity, email, phone number, address
- PKG can be off-line while registration is finished
- Extract slices *msk* to many secret keys w.r.t. identities; secret keys are delegatable

©Yu Chen, CAS  
18 / 67



©Yu Chen, CAS  
20 / 67

有没有什么问题？（没有）。这个还都是比较经典的，到后面可能就更绕一些了。

提问：这个跟签名有什么不一样？

陈宇博士：完全不一样。不管签名多少次，签名和加密是两个不同的。

提问：这个是不需要知道其他人的私钥。

陈宇博士：我加密的时候都不需要知道其他人的私钥，我加密的目的是什么？

为一个数据做一个访问控制对吧。现在就说我把这个加密数据放在这，我脑子里想一想，我希望哪些人有能力打开它，我在做这个加密之前想一下，想好之后放在那里，你只要有相应的属性就可以解开。你可以理解为脚本，但这个访问控制是由密钥和它共同控制的。

提问：其实是把访问策略作为加的一部分。

陈宇博士：对。

提问：这个访问策略的体系是由比特币的密码算法保证的？还是算法之上的？

陈宇博士：这肯定是密码算法本身保证的，所有的加密技术都相当于做自主访问控制，不需要任何的第三方平台，所有的控制权全部在自己手里，这个就是加密的魅力。

提问：怎么证明你的人是大于 18 岁啊？

陈宇博士：很简单，这个是个很好的问题。它要先到 PKG，我说我 18 岁了，

系里也看你是了，我说我是 CS 系的，给你一个相应的密钥，在这里。

提问：PKG 就是一个中心是吗？

陈宇博士：对。但这个访问控制，其实跟这个中心是没有关系的，这个东西可以是放在任何一个地方。它是由秘文本身实施的一个访问控制，它自带的一个访问控制功能，这就是加密。

提问：所以 PKG 是私钥？

陈宇博士：这里面抽取出的都是私钥。

你给我这个相应的 attributes，我给你相应的私钥，这个密钥写出来，可能就是一个 Zp 上的元素或者是一些组合，它本身不带有任何的语音系列，和 PKG 打个交道以后，所有的都是代数上的关系。这个不是随机的，这个是私钥，我不需要公开，这个是拿在自己手里的，不需要给别人看。

这就是公钥加密的一个极大的泛化，现在先退一步，公外加密的时候是用 PK，这个 PK 是大家都可以看到，这个也是。

提问：这个策略是每个人都一样还是每个人不一样？

陈宇博士：你想让谁看就写成什么样，传统的 PK，你想让谁看谁，是不是要把对方的 PK 填进来，现在你想让谁进来，就把策略填进去就可以了。

提问：这个是一一对应的吗？

陈宇博士：不是，这个是你只要符合就可以了，这里面有一个，不一定是 CS

才可以，我核定 18 进也可以。

提问：怎么样保证 PKG 是安全的？

陈宇博士：这个就是其他的了，你总要有一个信仰。

提问：假设这个是 A，那个是 B，那这个 A 是怎么找到这个公钥？

陈宇博士：这里没有任何公钥。

提问：怎么找到这个 attributes。

陈宇博士：自己想写成什么样就写成什么样，首先你加密的目的是什么。

我们首先要知道我们用一种什么语言来通信，什么语言来交互，拿商秘来说，比如现在约定 SM9，它的算法是不是都公开的。那我现在给某人发一个消息的话，这个加密算法是公开的，MPK 是公开的，他的名字往里一填就可以了，消息放在这里就可以了，加密，结束。

比如说我要加密给一个聪明帅气，我就写“聪明帅气”这个人才能解密。比如说到某个机构刷脸，确实很帅，就给他一个相应的私钥就可以了。

因为访问控制都是你加密的时候有一个控制，我想让谁看，谁不能看。

提问：解密的时候，解密不是拿自己的私钥解，而是第三方。每个组有一个私钥，每个组有一个私钥。帅气一个私钥，计算机专业一个私钥，几个混合在一起，满足它的条件，就能解。然后有私钥的，根据属性来产生对应的。

陈宇博士：这个 IBE 可以看成 IBE 最简单的版本，就是说现在加密的访问控

制数，是不是相同。这个 ID 我加密过来的时候，Alice 放在这里面的一个 ID，这不也是一个访问控制结构吗，其实相当于只有拥有 ID 对应私钥的人，才可以解密。我换成这个访问控制结构的时候，就说只有满足这个访问控制，拥有满足这个访问控制属性私钥的人才能够解密。

提问：请你回到前面那一页，这和 PKI 有什么不一样？

陈宇博士：这个很重要，我现在不需要知道某个人的公钥是什么，我只要知道他叫什么就可以了。

我这里仔细说一下 PKG 有什么样的问题，为什么才要有一个 IBE。我现在介绍这个密码，从最简单到高级的演化过程，怎么样跟着用户需求在走。

提问：这个好处 PKG 不需要在线。

陈宇博士：PKI 也不需要，它还有一个很隐秘的好处，可能不太容易看到，其实这个 C 和 P 都可以不停地向下代理，我有一个 PKG 以后还可以向下代理，这一点也在 IBE 里面后面显得非常重要，在这里面可能不是太明显。

这个 PKG 的信任，我觉得也是很可以理解的。

提问：PKG 会有密钥托管的问题。

陈宇博士：其实有时候很希望有一个 PKG，只要是政府部门，他们很希望这种东西的存在。所以说为什么以前科技认证要搞中心服务，要求有一个主私钥能打开所有的内容看一看。

提问：不考虑密钥托管的问题？

陈宇博士：这里面巧妙在什么地方，实际上这个 PKG 不要理解为第三方，它就是 Alice 自己。

举个例子，我们一屋子人都是好朋友，我有一个好资料，我自己当成一个 PKG，我自己生成一个 MPK、MSK，我现在就可以把我想加密的，我想和大家分享的资料，用我自己的 MPK，我自己制定一个访问策略，上传到第三方比如百度云上，发一个微信告诉你们已经把这个资料放到百度云上加密过了，你们拿私钥去下吧。这个私钥是什么？就是我为你们每个人发过去的私钥。我自己很多时候就可以是 MPK、MSK。

提问：所以不存在私钥匿名的过程是吗？

陈宇博士：不存在。

提问：PKG 也有它的扩展性、负载。

陈宇博士：对。

我们继续往下讲。大家发现到现在为止所有的加密企业都有一个特点，就是 All or Nothing，有我这个私钥什么都看得到，没有的话什么都看不到。

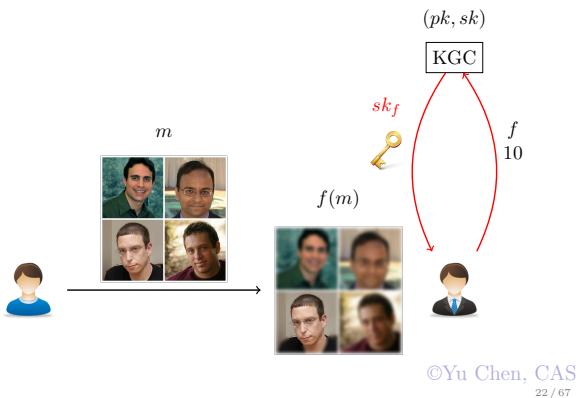
那么我们需要另外一种加密，我希望看到 All 和 Nothing 之间的一个状态。我们管他叫 Functional Encryption(FE)。

这个时候它的私钥，我可以走一个主私钥，SK 里面，去为任何一个函数  $f$  生成一个  $SK_f$ ，我解密的时候，比如我用  $PK$  对这个加密之后用  $SK_f$  去解，只能得到

$f(M)$ , 而不是  $M$ 。

我画图给大家解释。这里面  
已经把 PK 和 SK 画成个人了，这  
四个人我们照了一个合影，假设  
这个图是四宫格，我加密后这个  
图象已经看不到了。现在某个人

**Functional Encryption**  
Break “all-or-nothing” barrier



比如唐聪坐在第一排的，10 表示什么意思，第二行第一列，你是这个人吧，你就申请一个 10 相关的密钥，KGC 就把这个  $SK_{f, 10}$  给了。这个解密是什么样，如果他的解密结果，这样。传统的做法要么全部都看到，要么全部都看不到，要么是用一种很土的方法，把这四个逐块加密，这个就是一致性的。我通过给你不同的私钥，得到一个细粒度的访问控制，这个已经不是 all nothing，而是 all something。

提问：这里面可以任何组合吗？

**陈宇博士：** 这都是任意的函数。这个 FE 是什么？它实际上允许我们对一个加密的数据做一个有选择的计算，这个是已经远远超过传统的隐私保护的范畴。但它有一个问题，就是你做这种计算，其实这个计算实际上是一个解密的过程，你需要一个比较弱功能的  $SK_f$ 。还有一个更狠的想法，我能不能对加密的数据进行？根本不需要任何的 PKG。

刚才解密的过程，大家不要把它看得那么简单。关键是对它的理解上，对它最好的理解使用这个  $SK_f$ ，对加密的数据做一个计算，去掉之后是直接作用在  $M$  上了。（ $F$  可以任意构造）。

我们能不能对加密的数据进行一种公开的处理？也是可以的。同态加密，FHE，

它支持对加密后的数据做任意的操作。

举个例子，Alice 可以用自己的公钥和 SK 对一系列的消息 M1、M2、M3 加密得到 C1、C2、C3。这个人看到之后他不需要任何私密的信息，他可以任意选择一个函数 F，通过一个算法得到一个 C'，这个 C' 对应什么，只要有这个 SK 的话，对这个 C' 的解密就是 f(M1、M2、M3)，这就是同态加密。

提问：能不能举个加法同态的具体例子？

陈宇博士：可以。（现场白板）

提问：这个在数学上的一个例子，我想知道什么场合下能用到这个东西。

陈宇博士：这是后面讲的，你让举个例子，我以为是让讲这个。这个后面会有很具体的例子解释一下。

提问：这个是不是可以这样理解，Alice 有一串向量报给公钥进行加密，加密之后也是一串向量？

陈宇博士：不一定，向量加密之后明文和秘文差距非常大。我想说的是哪怕一组秘文都可以，它跟 m 是不是向量没有关系，因为一个私钥不可能只加密一个消息。

提问：就是多个消息做一个向量，多个秘文做一个向量。

陈宇博士：也可以。

提问：这里面关于同态的理解，在于可交换性，相当于我在原文做一个预算，加密之后的结果和先做运算再做解密的结果是一样。

陈宇博士：如果数学上的理解就更精确了。

提问：这个图是可以交换的。

陈宇博士：是。

这里面过一下 FHE 的发展和历史，这个问题 1978 年提出来的，能不能对数据做一个造拓，直到 1999 年，基于理想格，主策略是 SWHE+ 自主技术，加上对解密电路的一个混杂，但存在一个主要的问题，噪音增长过快，这有一定的噪音，噪音恰恰是安全性的根本，所以根本没有办法运营，这是第一代。

紧接着 2011 就有第二代，对噪音的控制做的更好。第三代是 2013 年，他们大大的缩减了公钥尺寸。之后对 FHE 做一个总结，因为它和 FE 的概念很像，共同点是远远超越了传统加密的范畴，更重要的是调和了这样一对看上去不可调和的矛盾。什么矛盾？第一个是我希望把一些隐私的数据外包，并且对这些数据计算，同时又希望保护数据的隐私。就是我又想要马跑得快，又不想要它吃草，但现在的应用相对比较少，这个也可

#### FE vs. FHE

以理解，因为它功能那么强大，怎么会做的简单，只有简单的东西，在功能实现上会有比较快。但我觉得至少不会等太久。

差别在什么地方？它的操作

##### Common

- both are cutting-edge of cryptography, go beyond traditional encryption
- still far from practical

##### Differences

- FE: manipulation requires  $sk_f$ , and the result is in plaintext
- FHE: manipulation can be done publicly, and the result is still encrypted

FE and FHE are suitable for different scenarios of  
*computing on encrypted data*

©Yu Chen, CAS  
26 / 67

对这个秘态数据的操作需要一个 **SKf**, 同时得到的结果是一个明文, 这是一个非常重要的区别。他们之间的区别使得他们在不同的应用场合有各自的应用。

提问：昨天我听了华为一个沙龙，里面有一个叫张风（音）的，安全保护的，他说他们现在跟百度在做相当于秘文的搜索，说他们已经做出来了，我想问一下您对这个事。

**陈宇博士：**密文搜索这一套技术应该相当的成熟，对称加密下的 **SSE**, 这都是非常成熟的技术。我知道你说的是可搜索加密，这部分应该比较简单。

提问：这个有什么区别？

**陈宇博士：**它（加密搜索）是它的一个极其特殊的一个特例，非常简单的一个特例。我前面提到的 **IBE**, 把它拧一下就是一个公钥加密的可搜索加密，结束了。

有的时候可以想一想这个问题为什么难做，为什么加密后的数据比较难检索？因为加密本身要把所有的语音信息全部破坏掉，所以没办法检索。这时候检索的时候，也是给它一个 **token**（音），可以很简单理解为做一个尝试的解密的匹配，匹配什么功能，没有什么神秘的。

可搜索加密，这应该是一个非常简单的应用。

更好的理解方式，你搜索也是对秘文做一个计算。

进入下面一个环节，**Key**

① Privacy of Data

② Privacy of Key

- ③ Privacy in Computing
  - Outsourced Computation
  - Multiparty Computation

④ Privacy in Proof

Privacy。这个大家关注的更少，刚才说的实际上是保护了消息的隐私，没有保护用的公钥的隐私。很多时候，我公钥的隐私也是非常重要的，比如比特币里就是这样，我加密给某个地址，我连这个地址都不想在加密后的明文里透出一点点，你看到的就是一个秘文，根本不知道我要给谁传情。这就是密码学里面的匿名性，加密后的，根本不泄露关于加密用的任何比特币的信息。

举个例子，你任意选两个公钥，  
**pk0** 和 **pk1**，我现在脑子里随便想一个 $\beta$ ，然后使用这个  $\text{pk}\beta$  对这个指定的消息加密，得到一个秘文 **C**。这种情况下对手都没有办法判定出，即使他知道这个消息是什么，他都没有办法

判定出这个秘文是在 **pk0** 下加密还是在 **pk1** 上加密，就说这个秘文本身不含有 **pk** 的任何信息，在计算的下面看不出来。

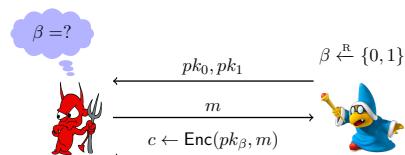
其实这就是一个很好的例子，看一下这个秘文本身这个 **gr1** 这部分跟 **pk** 没有任何的关系，根本没有用 **pk** 的信息，怎么可能漏出 **pk** 的比特信息。有的时候不是匿名的，很多时候比 **pk** 的信息作为秘文的一部分的时候，这就不是一个匿名的。

所有的密码学里算法和所有细节都是公开的，唯一保密的就是一个密钥。  
PKE 就是既提供数据保密，也提供公钥，它就是在 Zerocash 里有一个很好的应用。

提问：假如说我知道这个加密算法是什么，原文是什么，我就用 **pk0** 加密一

#### Basic Requirement - Anonymity

$c$  does not reveal any bit of  $pk$



- Anonymous PKE provides:  
data privacy + public key privacy  
used in ZeroCash
- Extends to IBE, ABE, PEKS

©Yu Chen, CAS  
29 / 67

下， $\text{pk1}$  加密一下，对照一下你的秘文是什么。

陈宇博士：只要达到语音安全，这个加密一定是，两次对同一个消息，同一个  $\text{pk}$  和同一个消息，两次加密的结果都是不一样的。

这个我觉得更多的是理解上的，在实际应用中，KGC 还要给用户一个  $\text{skf}$ ，就说我现在搜索的时候，都是把我的搜索能力带给一个程序，让这个程序对这个秘文进行检索。我给它实际上就是一个  $\text{SKf}$ ，我希望即使给了你这个能力检索，我希望你在极大的意义下，你拿到我这个  $\text{token}$ （音）之后不知道它对应哪个词，你拿到这个钥匙之后都不知道去开哪个门，这就是 **Private Key Privacy**，这个属于比较高级的应用，更多的是理论上的意义。

最后讲一下什么叫更强。就是抵抗更多强烈的攻击。之前讲所有的密码方案都是考虑一个非常理想的模型，假设这个  $f$  是一个密码算法的软硬件实现，它可以是一个线性算法，可以是一个解密算法。传统的模型都假设，我只能看到一个输入、输出，不知道内部的工作状态是什么样，也不知道内部所用的秘密信息、密钥是什么，这个是一个基本的假设。但是现实中五花八门的攻击战时出敌手做比这远远多的东西。

比如可以做电磁信号辐射，功耗分析、计时，声音分析（以色列做出来了，一个实验室就放了一个麦克风，看这里面空气振动，来看跑的是  $\text{pk0}$  还是  $\text{pk1}$  不知道是真假），有一个更遭的 MIT 的结果，现在的屋里比如那边在说话，那边一张白纸放在桌上，那边一个摄影机可以通过白纸的振动知道我讲的是什么，这个

也是万物互联嘛。

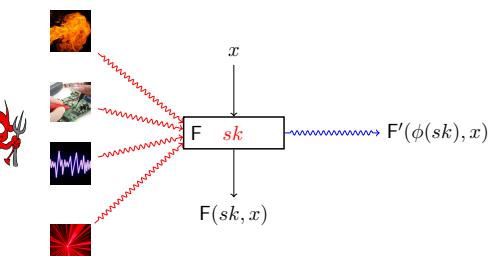
### Privacy Against Advanced Attacks

还有更狠的一点，拿  
火烤这个芯片，这是很可能的  
事情。搭线，或者把线切一下。

紫外线照射，激光打一打，它  
有可能干什么，或者把里面的  
逻辑都改了，得到一些额外的

信息。这是密码学在更强方面的研究，我们希望能研究出可证明安全的，就是对  
一揽子攻击可证明安全的密码方案，能抵抗许多攻击。

下面进入到 computing Privacy。有什么问题没有？



privacy against leakage & tampering attacks

©Yu Chen, CAS  
32 / 67

提问：硬件钱包，硬件外它的私钥是没有办法拿出来的。

陈宇博士：是没有办法拿出来，我可以去听啊，我不一定要搭建，我去让你  
运算一下的东西，我在边上听一下就可以了。

欧链科技：我补充一下，不知道在座有做硬件 key 的，就是硬件钱包。它主要的东西是把私钥烧到芯片里去了，计算的环境也是在芯片组里计算，但也分一代 key、二代 key，在成长过程中，也会有很多的方法包括测心跳的攻击，看它泄露出的消耗的能力、辐射，来猜测注入到里面的 key 那个私钥到底是什么，这个是实际发生过的，而且对于至少一代 key 里基本上防不了这个攻击，包括密码算法可以去看这样的攻击，或者有更强的密码算法。

陈宇博士：它的特点是不需要预先知道你是用什么方法来攻击我，我更希望

我设计出的天然可以抵抗所有这一类型的攻击，不需要针对某一种类型去打补丁，天生设计的算法就具有这种鉴别性。

### 下面讲 computing Privacy

更绕一点，我就讲的更直观一

① Privacy of Data

点。非常简单的对所谓的计算

② Privacy of Key

做一个理解，所谓的计算它的

③ Privacy in Computing

- Outsourced Computation
- Multiparty Computation

本质就是有一个程序，或者一

④ Privacy in Proof

个函数，有一个输入  $X$ ，得到一

©Yu Chen, CAS  
33 / 67

个  $Y$ 。这里面有一个比较重要的

理解，其实这个程序本身也是一种类型的数据。所谓的 computing Privacy，既要保护  $X$ 、也要保护  $Y$ 。

过去所有的数据和 program 都是同一个安全区域里，很多时候这种存储和计算本身都是在本地完成的，这个时候所谓的计算隐私并没有很强烈的需求。但是当这种分布式计算兴起之后，现在的数据和程序往往是分离的，不在一个地方。并且它的存储和计算有可能你都不知道发生在云端或者物端的什么地方，这个时候保护问题凸显出来。

一个很重要的问题，Privacy 在计算环境里考虑隐私是一个比较重要并且复杂的问题。为什么呢？首先我们看第一种类型的计算，外包计算里的隐私问题。它有什么特点，这个计算的本身往往是一个比较复杂的过程，或者是一个比较需要精巧结构的地方。第一个是我没有能力运算，比如我机器的性能不行，自己算不了。第二种我不会写相应的程序，我脑子笨编不出相应的程序，需要委托给别人，这就是它两个最重要的背景，都很重要。

举个例子，现在一个终端的设备，把这个  $X$  输入，通过一个 computing Privacy，把这个计算  $f$  外包到一个强力的服务器上，我们希望达到一个最基本的目标，协议执行完之后服务器不知道  $X$  是什么，客户端又拿到了相应的结果。

这里面有两种安全需求，客户端只关心他自己输入的这一部分隐私，他不关心  $F(x)$ ，甚至有时候希望服务器根据计算出的结果做相应的进一步的处理，这是第一种应用场景。第二种客户端既关系  $x$  的隐私，也关心  $F(x)$  的隐私，就说我既让你给我计算  $F(x)$ ，同时你访问出的结果，我也不希望服务器知道。服务器这边也有要求，他根本不希望你知道这个  $F$  算法是什么。所以说这就对应了两种分别的利用需求。

下面介绍 FE 和 FHE 解决需求 1 和需求 2。第一个是支持隐私保护的数据分  
析，Alice 自己有一对  $pk$ 、 $sk$ ，做加密得到  $C$ ，我上传到云端，同时希望 Bob，给他一个相应的函数， $F(y)$ ，生成密钥  $SKFy$  我希望他为我去做计算，这个  $C$  是一个谁都可以去下载的。他拿到这个  $SKF$  之后能得到  $F(y)$ 。

这个  $F(y)$ ，比如说两个点之间的距离，求两个向两之间的加速度。还有可以用来求均值、排序等等，它可以做一系列的统计工具。这个  $F$  可以是任意类型。还有简单的比较  $Fy(x)$ ，返回的结果  $x$  是否和  $y$  匹配，这就是用搜索加密，是还是不是。经过加密之后 Alice 得到他想要的结果，Bob 只能看到这个结果，并不知道这个  $Fy(x)$  是什么，客户端。

提问：他如果有计算的需求，可以把这个东西发给很多人去算，但计算的人其实是拿不到有效数据的。

陈宇博士：他只能得到结果，但不知道  $X$  是什么。就像前面的图一样，只能

看到我想要你看到的东西，其他地方你看不到，这个是我可以控制的。

这个里面专门提到了，他只关心我输入的隐私，不关心我输出的隐私。

提问：你泄露  $Y$  就是泄露  $X$ ,  $F$  得定义好，包括举例子内积，你泄露内积了，又泄露  $Y$ ，可以去尝试一下  $Y$ 。

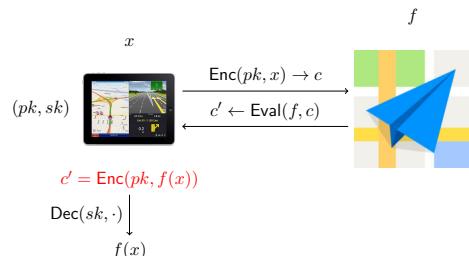
陈宇博士：这个时候他是希望服务器知道结果的，因为他希望你做进一步的操作。比如说我在一个网关上部署一个邮件检索的话，可以让服务器直接丢弃掉，我希望服务器知道处理的结果是什么。

提问：这个情况下  $X$  是公开的。

陈宇博士：对，这个是有应用需求的。

下面讲第二个需求，希望保护我输入和得到输出的隐私，同时这个 server，希望把计算的逻辑给隐藏起来。举个例子，导航，我们日常导航，如果是领导人的导航，他可不希望让高德的地图知道我从哪里来，要到哪里去。高德也不想说我把这个导航程序直接送给领导人，它最重要的就是那套导航的算法。

Privacy-Preserving Computing via FHE



©Yu Chen, CAS  
37 / 67

假如  $X$  里的信息是出发地和目的地，我把  $X$  用自己的 PK 加密以后发送给高德地图，高德地图这个  $F$  是导航算法，它对秘文作用一下，得到  $C$ ，这个  $C'$  得到导航路径，自己有私钥，解密出来，结束。

提问：相当于任何的  $F$  都可以。就这个还是一个畅想。

陈宇博士：你只要容忍一个小时之后才告诉我，我才把这个导航路径算出来，我觉得也是有可能，你只要愿意等。

这里面发现一个什么问题，在 **Outsource** 里，计算的参与方地位都不是对等的，都有一个外包的关系。很多的时候，我们希望这种计算的参与方，它的地位是对等的，这就是多方安全计算。

多方安全计算这个问题是姚先生 1982 年通过百万富翁的例子让大家都知道的，最开始是一个两方计算。两个富豪在餐厅里碰上了，都想证明比对方有钱，但又不想让对方知道自己钱到底有多少。简单的方法可以运行一个两方的计算协议，最后大家都知道谁的钱更多，都相信，而且又不知道对方具体的钱是多少，这个看起来简直是不可思议的事。实际上理论上解决的问题，都是这种看上去不可能有解决的问题，提供了一种及其巧妙的解决方案。

多方安全计算，就是使两个相互猜疑的交互方，有各自的  $X$ 、 $Y$ ，两个互相看不惯还要干一件事， $F(x)$  和  $F(y)$ 。**Alice** 和 **Bob** 抽象出通过 **MPC** 协议进行交互，协议完成之后两方都知道  $F(x)$  和  $F(y)$  是多少。这里面最重要的事，你在做一件事之前，你一定要知道我希望达成什么样的目标，不能打哪指哪，一定要指哪打哪。

这里面直观的感觉，这个多方安全计算，肯定是说协议执行完之后一定要隐藏起来，不能协议执行完，我的  $X$  就漏给别人了，实际上不是。因为这里面有一个很强的要求，这个  $F$  可以是任意的函数。那我们所能达到的，比如这个  $F$  可以就是简单的加，我们两方做一个简单的加法计算， $F=X+Y$ 。所以对对方安全的定

义，这种定义是错误的。正确的定义就算这两个人在交互执行完之后，得到的信息不会比这个结果得到的多更多，这是我们所能达到的最好的安全性。

学术研究方面肯定有这种想法，一定是执行完之后还要保密，不是，这个跟  $F$  紧密相关的，跟函数有关的，你们要计算一个什么函数，它所能达到的保护值是你不能知道更多，就行了。这个跟刚才举例子也蛮像的，有的时候跟  $F$  有关系。

这个例子概念大家没有问题吧，后面稍微有一些麻烦，我可能会跳的比较快一点。

这个是最杀手性的应用，好像在《自然》上都登了，DNA 比对，两个人 DNA 比对一下，相似度多少，同时双方又不想让别人知道我的基因图谱到底是什么，这个是最最杀手级的应用。

姚的做法，我讲一个最简单版本的一个例子。假设我现在做一个语音运算或者货运算，那就是一个逻辑门。这个逻辑门有两个输入， $x_0, x_1$ ，得到一个  $y$ ，这个电路我叫做  $C$ 。第一位有可能是 0，有可能是 1。这个线上的输入也有可能是 0，也有可能是 1。我现在怎么做？第一步，我对这个线路做一个杂化，我翻译成杂化，把整个计算过程进行编码，对计算逻辑进行编码，得到一个  $C$  带一个尖，还有个相应的标签的标。这个不要理解为一个物理上的，它只是一个形式，其实里面有一些内在的逻辑。我希望杂化以后达到什么样的结果呢？第一正确性，要求  $C(x) = C(k_s)$ 。对应标签，得到的所有信息，不会比你计算出的结果多更多。这个和 MPC 的安全应用是契合的，你看到的结果不会比你计算出的结果多。它本身单独并不能完成多方安全计算，还有另外一个很神奇的工具，Oblivious transfer，交互完希望达到什么样的结果呢，现在有三个消息，想通过这个协议把其中的一个消息发送给 Bob，Bob 只能拿到一个消息，我希望什么结

果呢？就说这个 `receiver` 只能拿到 `M0` 和 `M1` 之中的一个结果，当然这个结果可以由他指定，同时这三个人根本不知道 `receivev` 拿到的是哪一个消息。

现在看一下怎么用 `GC` 和 `OT` 做 `MPC`。第一步，我把这个函数，首先把  $F(x)$  写成这种形式，把输入固化到线轴里，得到一个新的线路。然后我对这个  $C$  做杂化，得到  $C$  尖和  $K$ 。首先这个  $C$  可以发给你，但不能把全部的 ( $C$  尖、 $K$ ) 发给你。最终得到编码以后的一个序列，同时这个 `Alice` 不能知道 `Bob` 得到了哪些，这个工作完成了，拿到  $K(y)$  之后得到结果，就应该是  $F(x, y)$ 。首先 `Alice` 肯定不知道 `Bob` 拿到了什么，`Bob` 计算出看到的信息不会比  $F(x, y)$  更多，最后把这个发回给 `Alice`，这个确实看上去比较难，这是一个最简单的版本。

我们现在都是介绍最简化的版本，假设所有 `Alice` 和 `Bob` 都是半诚实的，他们都必须照着这个协议走，后面会讲一些更复杂的东西。更简单的一个方法是用全动态做这个问题，他们的目标是  $x, y$ ，共同计算出  $F(x, y)$ ，`Alice` 加密，把秘文发给 `Bob`，`Bob` 用  $F(x, y)$ ，得到一个  $C'$ 。他自己用 `SK` 去解，再返回给 `Alice`。

最后讲一下实际世界中 `MPC` 是什么样的。

我也介绍一下这方面做的最好的几组人，第一个是 `Yehuda Lindell` 和 `Benny Applebaum`，都是大卫的学生。全动态方面，是这三个人，这两位都是 `IBM` 的，这位是 `MID`。

最后是讲一下交互式证明。有什么问题没有？

提问：刚才那个应用，为什么称它是杀手级应用？

陈宇博士：我为了吸引大家的注意。

提问：哪一年的应用？

陈宇博士：应该非常近，就是这一两年吧。

我们刚才已经说了，我们去做一个动态链计算，FHE 只是其中一种。就说这个 F 函数，其实这里面会有一个杂化嘛，MPC 里，这里是对 F，它可能对这些杂化做一些特定的优化，让你算的更快一些。

证明的隐私，大家可能在不同的场合听过很多遍，我以我的角度讲一下。密码学中的证明，今天想一想，什么是一个数学证明。一个数学证明其实就是一个数学证明。一个数学证明其实，你可以看成一串符号，这一串就是一些逻辑序列，它有一些公理、假设组成，最终推导出一个具体的结果。举个最简单的例子，我要证明一个定理， $N$  是一个合数，那我们  $T$  就可以是 3 和 5 是素数， $TN$  就是简单的逻辑，证明完毕。

从上面一个简单的例子我们观察到数学证明有以下几个特点：一它是非交互式的，是一个静态的过程，写在纸上你看就可以了，或者说有时候读数学定理有没有感觉有人跟你说话？没有，所以它是一个非交互式的，可以写在纸上。第二是非对称性的，体现在什么地方，很多的数学性需要花很长时间想出这个证明，但你一看原来是这样，大家都有这种，你验证这个证明式正确会非常快。第三，它不是一个零知识，证明总会泄露某些比较精巧的地方，比如你不看到证明，自己可能写不出来，你看过总要获得一些什么，它要泄露一些东西，有受收获。

① Privacy of Data

② Privacy of Key

③ Privacy in Computing  
• Outsourced Computation  
• Multiparty Computation

④ Privacy in Proof

©Yu Chen, CAS

46 / 67

大家都知道孪生素数的猜想，这个假设可能两百年了，他自己花了几十年去找到一个证明，但这个收稿日期 2013 年 4 月 13 号，一个月之后接受了，说明验证者花了一个月，其实实际上更短。他花了十几年想到一个证明，另外一个验证者，马上就说，确实是对的，这就是刚才说的不对称性。同时这个人他也知道怎么证了，它不是一个零知识，这就是传统证明的一些特征。

但现实社会中，很多证明其实是一个交互的过程。比如 P 断言，会问你一个问题，直到他自己确信这个断言确实是对的。比如在法庭上辩论，律师希望证明这个被告人是无罪的，就会有一些法官不停地问他一些问题，然后我之后回答的，律师之后回答都很好的话，法律就确认你是无罪的，这是很简单的例子，这也是证明。

还有大忽悠，赵本山对范伟说你有毛病，范伟说我没毛病，没事走两步，这就是问你几个问题，他确信了我真就毛病，这个其实就是一种交互的力量。

这些简单的例子告诉我们一个什么事实，证明一旦有交互之后，它要比非交互的证明，威力更强大。这个是学术上的一个结果，上世纪八十年代提出了交互式特别是零知识的证明性这个概念，他们因此得到的第一届哥德尔奖。

STOC 1985: The Knowledge Complexity of Interactive Proof Systems



Figure: Shafi Goldwasser & Silvio Micali & Charlie Rackoff

#### Interactive (Zero-Knowledge) Proof Systems

- First Gödel prize
- 2012 Turing award

©Yu Chen, CAS

51 / 67

下面开始稍微正式一点了，

证明到底是什么样，这个 X 可以理解为所有断言的集合，这个断言我们正确的才称为是一个定理，否则的话就是一个错误的结果。P 就想向 X 证明，中间可能经过一系列的交互，最终经过若干轮交互之后，V 根据交互的历史，输出、接受或

者拒绝，这里面  $V$  自己要有一个随机数，不能问你同样的问题对吧。

这里面有一个最重要的性质，要求以计算的角度，一定要求  $V$  是概率里的算法，这个算法要求什么，不管证明怎么产生的， $P$  怎么向我生成这个证明，我一定希望在多项式时间里验证这个证明是对还是不对，这个断言是真还是假，一定要快速的判断，这个是最重要的性质。

说证明系统一般都有三个性质，第一是完备性，这个合理性可以看成所有用的系统都必须的，这个是逻辑的自下性，只要是真的，最终都会接受。这里稍微说一下， $P$  能在很短的时间里，能为一个真实的命题，产生一个证明，当然它可能需要一些帮助，这个帮助是什么？这个帮助是一个  $X$  属于  $Y$  的证据。

只是说正常的  $P$  和  $V$ ，按照这个协议走，最后肯定能达到你所希望的效果，现在到这里为止没有说安全性任何的问题。那么安全性实际上说的什么？就说如果要证明给我的这个事实是错误的话，那你这个  $P$ ，其实有无穷多的计算能力，都没有办法让这个  $V$  以一个不可忽略的概率去输出 1，即使你骗术再高超都没办法骗我，当然允许有一小的概率，比如骗一万次我相信一次。

**Soundess** 保护的是弱势者，如果欺骗了  $x$ ，不管你怎么样绕开这个协议，最终都没有办法让我接受一个错误的断言。

提问：一个说没有假阴性，一个说没有假阳性。

**陈宇博士：**就说你说这个事实是对的话，我一定接受；如果不对的话，你骗不了我。

我们通常在学术里，修时候也可以把这个  $Pr$  走一个任意的无穷计算的能力，这时候得到一个 **arguement**。刚才所有的证明形式都是证明一个断言，就是证明

一个事实，这个证明的信息量还是蛮少的。很多时候向对方证明我拥有一个惊天大秘密，我拥有一个秘密，还不想告诉你，但又想让你知道我确实有一个秘密，这个是比证明更高一点，叫知识的证明。

我的证明过程需要证明我确实知道为什么  $x$  属于  $L$ 。比如我想向你证明，我给你一个  $h$ ，我告诉你知道它的离散对数，但又不知道这个离散对数什么。这个定义就不讲了，定义有点麻烦。

我们在现实世界里证明一个身份，Alice 说我是 CIA 特工，现实世界里是不是拿一个证件给你看一下你就相信了对不对。但在数字世界里我出示的是一个电子凭证，它可能有 CRN 我的签名，这个消息应该是 Alice 是一个 CRN agent，有一个签名。现在如果我一旦看了之后，我是不是也可以拿到这个文件到处去骗，我希望达到什么效果呢，我用名字去证明这件事。对这个断言的设计，可不能写成，Alice 是 CIA 这个消息，存在一个签名，不能这些写，这是一个废话。必须把这个断言写成我知道 sek for，我要知道什么，而这个 SIC 是保密的。

交互证明是一个两方协议， $P$  向  $V$  证明一个断言。现在说保证了  $V$  不受骗， $P$  也跳出来说我要受保护。通常这个证明有可能在证明过程中会泄露一些消息，但现在一个很有野心的想法， $V$  去一个很高的概率，相信  $x$  确实属于  $L$ ，同时不能知道为什么。跟这个 MPC，只

#### Motivation of Zero-Knowledge

让你知道这个结果就可以了，你  
不要知道为什么。

Soundness protect  $V$  from being cheated, how to protect  $P$ ?  
Usually, proof may reveal information.

Ambitious Goal:  $V$  ascertains with high probability whether  
 $x \in L$  without learning anything about why.

好比我上课，虽然不明白，  
但我觉得很厉害，就要达到这个  
效果。（已经达到了）。



*How to formally define "learn nothing but validity"?*

©Yu Chen, CAS  
56 / 67

最重要的是怎么样定义这个有效性。看一个简单的例子，数学上的定义，或者理论上的定义，这个  $x$  没有利用  $P$  的任何信息，所有看到的视角都可以自己模拟出， $V$  自己就可以编造这个世界，不需要  $P$  的参与。这也是一种精确的定义，其它的东西都看不到。

这里面我需要强调一个比较重要的东西，就是在零知识具体做应用的时候，大家都会认为， $CK$  实际上能保证的仅仅是什么，仅仅能保证这个证明本身不去泄露  $witness$  的消息。但这个断言本身，它有可能去泄露这个  $witenss$  的消息。所以说在应用零知识证明的时候，最麻烦的一点是你要很精巧的设计这个断言，你要告诉别人你到底要证明什么，这个是最难的一部分，如果不知道的话那就是用很强大的工具设计出一个漏洞百出的方案。

我再给大家写一下。（公式）所有关于  $x$  的信息会暴露在  $\pi$ 、 $H$ ， $ZK$  只能保证  $\pi$  这一部分不会额外的泄露，除了  $H$  以外，它不能保证这个本身的泄露，所以你在设计的时候一定要非常仔细。用  $ZK$ ， $ZK$  本身有很多的线程，关键是怎么去用它，这个是最难的。

给大家举个最简单的例子，讲零证明不需要任何的数学知识，可能大家在其它场合听过很多遍了。你怎么向一个红绿色盲证明两个球，确实一个是红色、绿色的球，首先他自己肯定看不到。我证明之后我又不想让他知道，本来他看不出哪个红绿，我也不希望他后来知道哪个是红哪个是绿，我只需要他知道这确实是一红一绿。

假设我是色盲，你们要向我证明，我现在把这两个球抓到手，反正我抓到手也看不出来，我背到后面，我脑子里随便想一个  $\beta$ ，如果  $\beta$  是 0 的话，我不交换，拿出来给你。如果  $\beta$  是 1 的话，我换一下手给你看。我问你问题，问这个证明者

问题，我有没有换手。如果这两个球确实是不同颜色的话，你每次都能准确地告诉我我有没有管手，都能准确地告诉我脑子里想的 $\beta$ 是什么。这就是完备性，只要你断言是真的，我会确信。如果我这个是假的话，我换不换手，你看上去没有任何的分别，只能以  $1/2$  的概率猜测我脑子里想的到底是什么，我只要重复几次，你肯定会算错。

类似的问题很多，大家可以看到那种美食节目，比如什么煮饭仙人啊，手艺多高超，真正检验一道美食的话，就是盲评。或者能喝出酒的不同，最好的证明办法就是盲评。

首先断言是真的话， $P$  肯定每次都能成功。如果断言是假的话，即使  $P$  有无穷多的能力，每次猜中  $1/2$ ，重复  $N$  次，都答对的概率是  $2$  的  $n$  次方分之一。

这个时候  $V$  本身是没有能力去验证这个断言本身的，这个证明本身是一个零知识的。为什么？你想  $V$  每次得到的信息是什么，得到的信息是你脑子里自己想的信息，你想你能得到其它额外更多的信息吗？不可能。

我要讲阿里巴巴四十大盗的问题，那个问题更复杂一些。这个交互式证明的威力来自三个地方，第一个是交互，第二个是每次都是随机决定是不是换手给你看。第三个允许小小的误差，有可能以极低的概率骗过我。

上面三个要素的构成决定了交互式证明极大的威力。一个很重要的问题，哪些语言有零知识证明？这个是什么意思呢？跟应用贴一下，哪些个断言我可以证明，它不是一个万金油，也是有限制的，这个东西写成什么样，能有一个相应的零知识证明。

其实这也不是一个问题，FOCS1986 的结果，所有的 NP 知识，都有一个 ZK。

刚才讲的是交互式证明，很多应用里不希望这个证明有任何的交互，比如一

个数学家在旅行过程中想出一个很美妙的证明想给同行炫耀，但他在旅游中把这个寄给同行之后就不能再交互了，因为他在旅游中，地址不确定，这时候需要证明是非交互的。这么简单的东西，把交互去掉是有极高的代价，需要引入一个第三方生成一个公共参考串，这是一个很重要的对应用上的限制。

1988 年的时候，其实对于 NIZK 来说只要有单向置换来说，对所有的 NIZK 都有一个 NIZK 的证明。后来差不多的时候，有两个学生在论文里有一个很简单的想法，只要有一个 public Coin ZKP 协议，就扭转为一个 NIZK。第一个把交互去掉了，第二个得到一个迅速，第三个把 CRS 都去掉了，但引入一个副结果，需要把那个函数看成一个随机，一般认为它不是现实中存在的，总是有那么一点点担心。

现在 NIZK 应用中的瓶颈，假设我想证明这个断言是什么，比如说我知道这个  $w$ ，使得这个  $F(w) = x$ ， $x$  说我不想这个  $w$  告诉大家，这个  $w$  可以理解为比特币的钱包，比特币里面具体的金额。

目前 NIZK 生成的证明还有验证的开销，不仅跟断言的长度有关，而且跟这个断言本身定义的函数是有关系的，验证这个函数的时间是有关系的。实际当中，这个  $F$  运算的时间，要远远的大于这个线路的程序的大小。

就是这一点导致了传统的 NIZK 在现实中运用并不是那么方便，甚至是一种很昂贵的操作。

这两个人提出一个新概念，把这个 S、N、A、R、K 大写，ZKSNAK。这里 P 它跟  $F$  的运算时间没有关系了，只跟大小有关，这是一个极大的下降。

我们怎么样把一个 NIZK 变成一个 ZK-SNAK？要借助一个 PCP 定理。PCP 就是一个概率可验证的证明。它的特点是这个证明可以被多线路线程使用，并且仅访问部分的比特，比如 QN 比特，就可以以  $1/2$  完成验。

举个例子，它其实相当于什么，现在你把这个证明发过来之后，我不需要检查证明所有的比特，我只要看几位就可以知道，以很大的概率判断这个证明对不对，相当于什么，相当于把传统 **NIZK** 的证明改写了一种方式，使得如果你这个证明是一个错误定理的证明的话，我会让你一点点微小的错误，弥散到整个证明几乎处处都是。好比果酱抹的很均匀，错误弥散的到处都是。

其实 **PCP** 定理告诉我们什么，所有的 **NP** 语言都有一个  $O(\log n)$  的证明。也就是我只检查常数的比特就完成了证明的验证，**PCP** 定理也是之后最重要的一个成果。

现实中这个 **SNARK** 有两种途径，**NIZK+RO** 下的，还有是 **NIZK+ris**，这两个构造途径的共性，正方还是需要很多的时间去产生这个证明。它虽然不好但也有一个好处，就是不需要 **trapdoor**，但这里面需要你相信这个参数是真的，没有被人做手脚。

提问：这个被泄露出来的话，**crs**。

陈宇博士：本身你泄露出来就是让大家知道，关键是生成的时候在线。

提问：怎么生成的这个事情要保密是吗？

陈宇博士：不是要保密，是要按照你的协议，按照正常的算法去生成，不能恶意的生成。这个也是一个很麻烦的，现在 **ZK** 是随便找六个人共同生成这个 **crs**，为了保证它没有被做手脚。

欧链科技：**ZK** 当时发布的时候，当时找了六个比较出名的密码专家，然后现

场销毁了，为了保证 ZK 的团队也不能从中作恶，保证了这件事。

提问：crs 可以做什么？

陈宇博士：想干什么干什么，相当于是上帝视角。

这个也是成功用在 ZeroCash 里面。

他其实是以色列理论密码机非常伟大的科学家，并且把他那做套推向了实际而且也推导了极致。这个人是他的学生，他也很厉害。

谢谢大家。

欧链科技：感谢陈博士为我们做了一个长达两个小时的报告，其实我觉得在整个的报告中，可能大家最感兴趣的，和之前听了不一样的地方，就是我们对于隐私的这个理解，可能和我们之前很多地方跟不一样。密码学家是从什么样的角度考虑隐私问题，怎么样做定义，可能陈博士给了一个全新的理解，我们密码学里面讲的隐私和大家理解的隐私还是很不一样的。

下面大家有什么问题，可以做一个简单的交流。

提问：拿一种私钥去签一个证明，有一个问题是如果签的越多，这个被证明的可能性越大。

陈宇博士：拿比特币证明，每个人有一个  $pk$  和  $sk$ ,  $CKP$  丢掉是最糟糕的事，即使这个  $CKP$  没有丢掉都可以伪造出一个签名，我看到了一系列的信息和对应的签名， $M_1$  一直到  $i$ ，我看到有一天突然找到规律了，我可以为一个  $M_k$  生成一个合法的签名，这时候我已经对系统造成很严重的危害了，你说的是这种事情吧。

但实际上我们密码学里，它要满足的一个最弱的安全性，其实你看到无穷多个，这种消息对等，你都没有办法伪造出一个新的，不可能同样伪造出一个合法的无穷多，这个无穷多跟你的计算能力有关， $N$  可以选择非常大，都没有办法造出。

提问：比特币有一些实践中，有的叫一币，每次都换一个新的。

陈宇博士：那个可能要仔细的考察一下，它的应用背景，是不是在一些很特殊的货币里需要。因为它不是由于安全原因不能用，而是因为其它的原因，它只用一次就可以了，不会用第二次。所以为了迎合这种需求，理论上我看到的更多，我肯定知道的更多。你知道的更多是那个信息下你知道的更多，但现在考虑的都是计算复杂度，你只要算出来就可以了，你知道那么多知识对你没有用。

欧链科技：我补充一下，我们在比特币的应用里，强调每次应用的时候要申请一个公私钥对，这是从密钥管理的校对出发的，实际上怎么样保管好你的私钥是一件很困难的事。陈博士前面讲的是公开的这部分，算法之间保证了这个安全，但是私钥的安全保护，又属于另外一个范畴了，所以我们也是建议私钥不要使用很多很多次，使用很多次的时候可能在算法层面上不会破解，但它可能被旁边的人偷看到或者被监听到，可能是出于这种考虑。

提问：我想问 ZK，比特币上各种交易都是有限的，要证明是有限的。要证明，生成这个证明很难，难在什么地方，这个集合是有限的，可以事先先产生的。

陈宇博士：每次生成的断言是不一样的，每一次交易都要给一个证明，每次

证明的内容是不一样的，那不就结了嘛。

提问：比如我要证明我有多少钱，大于多少钱，这个是普遍在用的。

陈宇博士：形式是这样，但每次的内容会不会变，每个人证明的东西不一样。

今天有那么多钱，今天的钱多了，你证明的内容是不是变了。

这个人说我的账户大于 100 块钱，那个人说我的账户大于 99 块钱，这个变化还是非常多的，这个命题本身还是，不是一个有限的集合。

为什么生成证明很难，最关键的是怎么写这个证明，可能很短一个证明，**NIZK** 的证明，可能一个 **NIZK** 两行就写成了，**PCP** 的证明可能写成一本书那么长，这个相当于特殊的格式去写证明，使得它具有这种，使得让你验证的很快。我让你验证快，我是有代价的，我要花一种很复杂的格式去写这个证明。其实很多时候是希望你验证方会快一些，通信代价少一些，有的时候不太关心自己算的生成这个证明有多慢，就在这个。

提问：而且这也是说验证也是个概率。

陈宇博士：这个是精髓，看一下这里，允许小小的误差。

提问：误差有多大？

陈宇博士：想控制在多小就控制在多小，可以串行的减少也可以并行的减少，都可以。如果想更精确一点，多嵌入的合成。当然这种合成也要非常小心，你得到的安全结果可能是完全不一样的，这都是几十年没有搞清的问题，你们最好有什么用什么。而且你多运行几轮，每次都要把这个证明再写一遍。不是说一个 **x**

一个 $\pi$ ，你只能走一轮，再走一轮可能再形成一个 $\pi$ 。

提问：我们基本上可以把这三个东西都可以用 ZK 标签。

陈宇博士：你要证明什么一定要想好，要把它设计的非常小巧，就是你怎么去设计你要证明什么。把那个 NP 断言给设计出来。

欧链科技：可能在 zerocash 里面，这个断言已经不是说我转了多少钱给谁，在这里面它的语言体系变了，我转了一笔钱给谁，这个数字是不告诉你的，这个钱加上余额是满足原先手上的钱。所以在设计的断言的时候，到底想要保护哪些内容，要重新的专门设计，这样才会是一个更好的方案。

陈宇博士：更重要的是你很多转账的交易，可能那个数字都是用同态算法加密过的，这个是保证你金额的隐私，这一个组件是干不了的，需要一系列的组件。你加密了这个信息，满足一个什么样的条件。

提问：现在出了 zerocash 之外还有什么应用场景？

欧链科技：我们看到比较出名的在 zerocash 这一个，但有可能在一些小的地方吧，可能会有一些，在尝试性的应用。

陈宇博士：只要是任何需要证明的，ZKsnark 都是最好的选择，你总是希望你算的很快，只要以前用 ZK 的地方都可以用 ZKsnark。

提问：之前同态加密这一块，比如加法同态，我能想到的，像你之前讲到的，可能在加密或者运算外包的时候有一些用处，在别的地方还会有实际的用处吗？

陈宇博士：比特币里面就是。

提问：具体的是什么场景呢？

陈宇博士：这个是别人告诉我的，不是我想出来的。因为我要自己想出来的话，我就直接就告诉你们了。

提问：大的范围还是想用在比特币交易当中做。

陈宇博士：有一个麻烦，我讲一下麻烦。如果我一旦把这个交易的金额隐藏的话，比如这个交易是 PK，比如 100 块钱，以这种形式广播出去的话，就没法记账了。你这个钱，经过大家去记账，你现在已经加密状态，大家怎么知道到底是多少钱，这就是需要有一定的秘态计算的能力，还要和 NIZK 等等全部绑在一起全部完成。

欧链科技：可能在未来一两个月，在某一条链上可以看到。

陈宇博士：应该是已经部署蛮长时间了。

提问：包括之前您讲的很多的数学的基础，更多的还是一个组合来实现一个具体的，所设定的安全的需求，就是我们要保护什么，我们要去做什么事情。其实我们考虑问题，更多还是应该从需求角度去想。

陈宇博士：而且还会有什么问题呢，比如说你有可能牵一发动全身，你采用什么样的加密算法，有的东西简单的加密解密，那都是即插即用的，但像比特币

涉及到很多复杂的交用，跟应用贴比较紧的时候，一定要一开始设计的比较清楚，有比较好的扩展性，不然的话后面部署可能要全变。因为如果这个加密算法设计不好的话，你会发现对这个秘态的数据没有任何的可操作性，没有办法证明，这个是最麻烦的。但是我以前做纯粹的密码理论，跟这个一点关系都没有。

提问：我想保护哪一部分数据可以专门的去设计。

陈宇博士：从头到尾一直在说，一定要搞清楚自己想要做什么，这个是很重要的，很精确的描述就是想什么，达到什么样的程度。

提问：其实更多的是说我们需要什么样的。

陈宇博士：对，总是要有的放矢，这个蛮重要的。

提问：站在更加科学或者更加贴近密码学的思维方式，我们需要什么样的需求，然后再做什么样的保护。

陈宇博士：是。

提问：MPC 现在有什么应用场景吗？

陈宇博士：我讲个哗众取宠的应用，还有杀手级的应用。就是 DNA 比对啊，各种很多。两个人在比特币上，两个大咖看不惯，比一下谁账户里钱多。

提问：我听说某公司用 MPC 做数据的共享，但他们没有做成功，不知道这个难度在哪里。

陈宇博士：MPC 的一个特点，我们有两个集合来求一个交集，只知道我们两个共同的是什么，其他的地方都不知道，这个也是 DNA 比对需要的地方。你们俩相同的地方，大家都知道我们俩都有，其他的地方各自保密。

欧链科技：其实从密码学的角度来说，MPC 会有各种各样的方案，包括可以实际应用的，或者是理论上有新的更好的，但更主要的是需要明确一下到底需要东西，应用实际是什么。刚才说数据共享，可能这个命题太大了，我们需要明确在数据处理中达到什么样的效果，再选择相应的密码方案，才能做出一个更好的东西出来。

提问：现在区块链跟密码学的前沿，发展到什么程度，现在有什么新的东西出来，您有研究吗？

陈宇博士：我最后一页那两个人，都是做理论相当顶尖的，他们把实际应用都做的那么好，这个是很少见的相互促进，就是为了有这个应用，把 ZKsnark 推到极致，以前不敢想象有这么快的速度。

欧链科技：我个人认为从功能实现上来说不是一个太大的问题，可能是整个团队里面会去支撑跨平台的编译可能还没有跟上，可能更多的精力放在其他地方。如果大家没有问题了，那我们今天就到这，再次感谢陈宇博士精彩的演讲。

【完】

OracleChain 团队

2017 年 11 月 25 日