DePIN科普文：IoTeX、DePHY和peaq等基础设施是怎么运转的

作者：Pika，Sui公链大使，DePIN研究员

## 导语：

尽管DePIN赛道在当下十分火热，但DePIN相关的物联网设备要大规模接入到区块链，仍存在技术上的障碍。一般而言，若要将物联网硬件接入到区块链中，要经历以下三个关键阶段：

1. 硬件设备的可信任运行；

2. 收集验证并提供数据；

3. 将数据分发到不同应用。

这三个阶段中存在不同的攻击场景与反制手段，需要引入各种机制设计。本文从项目工作流程与协议设计的角度，回顾分析了物联网设备从可信任产生数据，验证存储数据，通过计算产生证明，和向区块链rollup数据的整个流程。如果你是DePIN赛道的创业者，希望本文可以在方法论和技术设计上对你的项目发展有所帮助。

下文中，我们以空气质量检测的场景为例，结合IoTeX、DePHY、peaq这三个DePIN基础设施进行分析，向大家阐明DePIN基础设施是如何工作的。此类基础设施平台可以对接物联网设备与区块链/Web3设施，帮助项目方快速启动DePIN应用类项目。

硬件设备的可信任运行

硬件设备的可信任，包括设备身份的信任与程序执行可验证无篡改的信任。

### DePIN的基础工作模式

在大多数DePIN项目的激励方案里，硬件设备的运行者会对外提供服务，以此为筹码向激励系统索要奖励，比如在Helium中，网络热点设备通过提供信号覆盖，来获取HNT奖励。但在从系统中获取激励前，DePIN设备需要先出示证据，证明自己的确按要求付出了一定“努力”。

这类用于证明自己在现实世界中提供了某类服务、进行了某些活动的证明，被称为物理工作证明(Proof of Physical Work, PoPW)。在DePIN项目的协议设计中，物理工作证明占有举足轻重的地位，相应的也存在各种攻击场景与对应的反制手段。

DePIN项目要依托于区块链完成激励分发与代币分配。类似于传统公链中的公私钥体系，DePIN设备的身份核验流程中，也需要使用公私钥，私钥用于生成和签署“物理工作证明”，公钥则被外界用于验证上述证明，或作为硬件设备的身份标签(Device ID)。

除此之外，直接用设备的链上地址接收代币激励并不方便，因此DePIN项目方往往在链上部署一个智能合约，合约中记录着不同设备持有人的链上账户地址，类似于数据库中一对一或多对一的关系。这种方式下，链下物理设备应当收到的代币奖励，可以直接打到设备持有人的链上账户里。



### 女巫攻击

绝大多数提供激励机制的平台，都会遇到“女巫攻击”，就是说有人可能操控大量的账号或设备，或是生成不同的身份证明，伪装成多个人，拿多份奖励。以我们前面提到的空气质量检测为例，提供此服务的设备越多，系统分发出去的奖励也越多。有人可以通过技术手段，快速生成多份空气检测数据以及对应的设备签名，制造大量的物理工作证明来获利，这会使DePIN项目的代币陷入高通胀，所以要制止此类作弊行为。

所谓的反女巫，如果不采用KYC等破坏隐私性的方法，最常见的措施就是POW和POS，在比特币协议中，矿工要付出大量的算力资源，才能获得挖矿奖励，POS公链则直接让网络参与者质押大量的资产。

在DePIN领域中，反女巫可以归结为“抬高物理工作证明的生成成本”，由于物理工作证明的生成，依赖于有效的设备身份信息（私钥），所以只要抬高身份信息的获得成本，就可以防止某些低成本生成大量工作证明的作弊行为。

针对上述目标，一种相对有效的方案是，让DePIN设备生产商垄断身份信息的生成权限，对设备进行定制化处理，为每部设备录入唯一的身份标签。这就好比，由公安局统一记录全体公民的身份信息，只有在公安局数据库里可查的人，才有资格领取政府补贴。



（图片来源：DigKey）

在生产环节，DePIN设备厂商会使用程序在足够长的时间里生成根密钥，然后随机选择根密钥使用eFuse技术写入到芯片中。这里要科普下，eFuse（可编程电子熔断器）是在集成电路中存储信息的电子技术，录入的信息通常无法被篡改或擦除，具有较强的安全保障。

在这种生产流程下，设备持有者和生产商都无法获知设备的私钥，以及根密钥。硬件设备可以在TEE的隔离环境中，从根密钥导出并使用工作密钥，包含签署信息用的私钥，和交由外界验证设备身份的公钥。TEE环境外的人或程序都无法感知到密钥的细节。

上述模式下，如果你想获取代币激励，必须要从专属厂商那里购买设备。女巫攻击者若想绕开设备厂商，低成本生成大量的工作证明，就需要破解厂商的安全系统，将自己生成密钥的公钥注册到网络许可设备中去，女巫攻击者很难低成本发起攻击，除非设备生产厂商监守自盗。

而一旦人们发现设备厂商有作恶的可疑迹象，可以通过社会共识的方式对DePIN设备生产厂商进行曝光，这往往会使得DePIN项目本身连带遭殃。但多数情况下，设备厂商作为DePIN网络协议的核心受益方，大多没有作恶动机，因为让网络协议井然有序运转的话，卖矿机赚的钱会比DePIN挖矿赚的钱多，所以他们更多会倾向于不作恶。



（图片来源：Pintu Academy）

如果硬件设备不是由中心化生产商统一供应的，那么当任意一台设备接入DePIN网络时，系统要先确认该设备具有协议要求的特性。比如，系统会检查这些新加入的设备，有没有专属的硬件模块，没有此类模块的设备往往无法通过认证。而要让设备拥有上述硬件模块，要花费一定的资金，这就抬高了女巫攻击的成本，进而达到反女巫的目的。在这种情况下，还是正常运行设备而非制造女巫攻击更为明智和稳妥。

数据篡改攻击

让我们脑洞一下，如果某台设备收集到的空气质量检测数据，其波动性越强，系统就认为数据更有价值，并为此提供更多奖励，那么任何设备都有充足的动机伪造数据，让其故意表现出高波动性。即便是由中心化厂商做身份认证的设备，也可以在数据计算过程中“夹带私货”，对收集到的原始数据进行改写。

该如何保证DePIN设备是诚实可信的，没有对收集到的数据进行肆意修改呢？这需要用到可信固件(Trusted Firmware)技术，其中较为出名的是TEE(Trusted Execution Environment), 还有SPE(Secure Processing Environment).。这些硬件层面的技术，可以保障数据在设备上按照事先验证过的程序来执行，计算过程中没有“夹带私货”。



（图片来源：Trustonic）

这里简单介绍下，TEE(可信执行环境)通常是在处理器或处理器核心中实现的，用于保护敏感数据，执行敏感操作。TEE提供了一个受信任的执行环境，其中的代码和数据能够受到硬件级别的安全保障，以防止恶意软件、恶意攻击或未经授权的访问。比如, Leger, Keystone这些硬件钱包都有使用到TEE技术。

大多数现代芯片都支持TEE，尤其是针对移动设备、物联网设备和云服务的芯片。通常情况下，高性能处理器、安全芯片、智能手机SoC（系统级芯片）和云服务器芯片都会集成TEE技术，因为这些硬件涉及的应用场景，往往对安全性有较高的追求。

但是，不是所有的硬件都支持可信任固件，一些较低端的微控制器、传感器芯片和定制的嵌入式芯片可能缺乏对TEE的支持。对于这些低成本芯片，可以通过探针攻击等手段去获取芯片内留存的身份信息，进而伪造设备身份和行为。比如，攻击获取到芯片上保存的私钥数据，然后使用私钥对篡改或伪造的数据进行签名，伪装成设备自身运行生成的数据。

但探针攻击依赖于专门设备和精确的操作、数据分析流程，攻击成本过高，远高于从市场上直接获取这类低成本芯片的成本。相比于通过探针攻击等手段破解伪造低端设备的身份信息来获利，攻击者会更愿意直接购买更多台低成本的设备。

数据源攻击场景

前面提到的TEE可以保证硬件设备如实的生成数据结果，只能证明数据在输入到设备内部后，没有被恶意的处理，但无法确保数据在进行计算处理前，其输入源头可信，这其实类似于预言机协议面对的难题。

比如，某台空气质量检测仪，被放置在了排废气的工厂附近，但有人在夜里用密闭的玻璃罐把空气质量检测仪罩起来，那么这台空气检测仪获取的数据必定不真实。但上述攻击场景往往无利可图，攻击者大多数时候没必要这么做，因为是费力不讨好。对于DePIN网络协议而言，只要设备满足诚实可信的计算过程，付出了满足激励协议所要求的工作量，理论上就该获得奖励。

## 方案介绍

### IoTeX

IoTeX提供了W3bStream开发工具，将物联网设备接入到区块链和Web3当中。在W3bStream物联网端的SDK中, 包含了通信和消息传递、身份和凭证服务以及密码学服务等基本组件。



W3bStream的IoT SDK 对加密功能的开发非常完善，包含多种加密算法的实现，比如PSA Crypto API, Cryptographic primitives, Cryptographic services, HAL, Tooling, Root of Trust 等模块。

有了这些模块，可以在各种硬件设备上，用安全或欠安全的方式去对设备产生的数据进行签名，并通过网络传递到后续数据层供验证。

### DePHY

DePHY在物联网端提供了DID(Device ID)认证服务。DID由生产商铸造，每一个设备都有且仅有一个对应的DID。DID的元数据可以自定义，可以包含设备序列号、型号、保修信息等等。

对于支持TEE的硬件设备，最开始由生产商生成密钥对，使用eFuse将密钥写入芯片中，而DePHY的DID服务，可以帮助生产商根据设备公钥来生成DID. 而生产商生成的私钥除了被写入到物联网设备，就只有生产商持有。

由于可信任固件可以实现安全可靠的消息签名与硬件端私钥保密，如果人们发现网络中存在作弊生成设备私钥的行为，基本就可以认为是设备生产商在作恶，就可以溯源到对应的生产商身上，实现信任溯源。

DePHY的用户在买入设备后，可以获取设备的激活信息，再调用链上的激活合约，将硬件设备的DID与自己的链上地址关联绑定，进而接入到DePHY网络协议中。物联网设备经过DID设置流程后，就可以实现用户与设备之间的数据双向流动。



当用户通过链上账户向设备发送控制指令时，流程如下：

1. 确认用户拥有访问控制权限。由于设备的访问控制权限以metadata 的形式写在 DID 上，可以通过检查DID来确认权限；

2. 允许用户和设备开通私密通道建立联接来支持用户控制设备。DePHY relayer 除了 NoStr relay 外，还包含 peer-to-peer 的网络节点，可以支持点对点通道，由网络里的其他节点帮忙中继流量。可以支持用户在链下实时控制设备。

在物联网设备向区块链发送数据时，后续数据层会从DID上读取设备的许可状态，只有通过注册被许可的设备才能上传数据。比如被生产商注册过的设备。



这个DID服务另一项有意思的功能在于提供了物联网设备的功能特性(trait)认证。这项认证可以识别出物联网硬件设备是否具有某些特定功能，达到足以参与特定区块链网络的激励活动的资格。比如一台WiFi发射器，通过识别到具有LoRaWAN的功能(trait)，可以认为具有提供无线网络连接的作用，也就可以参与到Helium网络中。类似的，还有GPS trait, TEE trait等。

在拓展服务方面，DePHY的DID还支持参与质押，链接可编程钱包等，方便参与链上活动。

### peaq

### peaq的方案比较奇特，它的方案分为三个等级，分别是源自设备的认证、模式识别校验、基于预言机的认证。



1.源自设备的认证。peaq同样提供了生成密钥对，在设备上使用私钥签署信息，将设备地址 peaq ID 绑定用户地址等功能函数。但是，在他们的开源代码中却找不到可信固件的功能实现。peaq简单的使用私钥对设备信息进行签名的认证方式，并不能保证设备的诚信运行和数据未经篡改。peaq更像是一个乐观Rollup，默认设备不会作恶，然后在后续阶段去验证数据的可信状况。

2.模式识别校验。第二个方案是结合机器学习、模式识别。通过学习之前的数据得到模型，当新的数据输入时，与先前的模型做比较，判别是否可信。但统计模型其实只能识别出异常数据，并不能判断物联网设备是否诚实运行。

比如，城市A中的某台空气质量检测仪放置在了地下室，收集产生的数据与其他空气质量检测仪都不一样，但并不代表数据伪造，设备仍在诚实运行。另一方面，只要收益足够大，黑客们也愿意使用GAN等方法，生成机器学习难以鉴别的数据，尤其是判别模型公开共享的情况下。

3.基于预言机的认证。第三个方案是他们会挑选一些更受信任的数据源作为预言机，与其它DePIN设备收集上来的数据进行比较验证。比如，项目方在城市A部署了一台精确的空气质量检测仪，其他空气质量检测仪收集的数据如果偏差太大，就被认为不可信。

这种方式一方面给区块链引入并依赖权威，另一方面，也可能因为预言机数据源的采样偏差，而使得整个网络数据采样都出现偏差。

就目前的资料来看，peaq的基础设施，在物联网端无法保证设备和数据的可信任。（注：笔者查阅了peaq的官网、开发文档、Github仓库，以及一份仅有的2018年白皮书草稿。即使向开发团队发送邮件，也未能在发稿前得到更多补充说明资料）

## 数据的产生与发布（DA）

DePIN工作流程中第二个阶段主要是收集、验证物联网设备传递过来的数据，保存起来向后续阶段提供数据，要确保数据能完整无误、可被还原的发送给特定接收方，这被称为数据可用性层(DA层)。

物联网设备往往通过HTTP, MQTT等协议，将数据和签名认证等信息广播出去。而DePIN基础设施的数据层在接收到设备端的信息时，需要验证数据的可信度，把通过验证的数据汇集存储起来。

这里介绍下，MQTT(MQ Telemetry Transport)是一种轻量级的、开放的、基于发布/订阅模式的消息传输协议，旨在用于连接受限的设备，如传感器和嵌入式系统，在低带宽和不稳定网络环境下进行通信，非常适合物联网(IoT)应用程序。



在验证物联网设备消息的环节，会包含设备可信执行的认证和消息的认证。

设备可信执行的认证可以结合TEE.。TEE通过将数据收集代码隔离在设备的受保护区域中，确保了数据的安全收集。

另一种方式是零知识证明，这种方法使得设备能够证明其数据收集的准确性，同时又不泄露底层数据的细节。这种方案因设备而异，对于性能强大的设备，可以在本地生成ZKP，对于受限制的设备，则可以进行远程生成。

在认证了设备的信任之后，使用DID去验证消息签名，就可以确定消息是由该设备产生。

## 方案介绍

### IoTeX

### 在W3bStream中，分为可信任数据收集、验证，数据清洗，数据存储这三个部分。

可信数据的收集、验证使用了TEE和零知识证明的方法。

数据清洗是指将来自不同类型设备上传的数据格式统一化、标准化，便于存储和处理。

数据存储环节，允许不同应用项目通过配置存储适配器来选择不同的存储系统。



在当前的W3bStream实现中，不同的物联网设备可以直接把数据发送给W3bStream的服务终端，也可以先把数据通过服务器收集后，再发送给W3bStream的服务器终端。

在接收到传入的数据时，W3bStream会像一个中心分发调度器那样，将传入的数据分发给不同的程序去处理，而W3bStream生态内的DePIN项目，会在W3bStream上申请注册, 并定义事件触发逻辑(Event Strategy)和处理程序(Applet).



每部物联网设备都会有设备账号(device account), 归属于一个而且也只能有一个W3bStream上的项目。因此，当物联网设备的消息传递到W3bStream服务端口时，可以先根据注册绑定信息，重定向到某个项目，并验证数据的可信。

至于前面提到的事件触发逻辑，可以定义从HTTP API终端、MQTT话题接收到的数据信息，以及检测到区块链上的事件记录，检测到区块链高度等可以被触发的事件(Event triggers)类型，并且绑定对应的处理程序去处理。

处理程序(Applet)中定义了一个或多个执行函数，被编译成了WASM格式。数据的清洗和格式整理就可以通过Applet去执行。处理之后的数据被存放到由项目定义的key-value数据库中。



### DePHY

DePHY项目采用了更为去中心化的方式来处理和提供数据，他们称之为DePHY消息网络(DePHY Message network).



DePHY消息网络由无许可的DePHY中继节点(relayer)组成。物联网设备可以通过任意一个DePHY中继节点的RPC端口将数据传入，传入的数据会先调用中间件，结合DID去验证数据可信任。

通过信任验证的数据需要在不同的中继节点之间同步，形成共识。DePHY消息网络采用了NoStr协议来实现。NoStr原本的用途是用于构建去中心化的社交媒体，还记得之前有人用NoStr代替Twitter而大火么，用在DePIN数据的同步中居然也巧妙的合适。

在DePHY网络中，每台物联网设备存储的数据片段，都可以被组织成一棵Merkle树，节点间会彼此同步这棵Merkle树的root，以及整棵树的tree哈希。当某个Relayer得到上述Merkle Root和Tree哈希后，可以快速定位还缺少哪些数据，方便从其他Relayer中获取补齐。这种方法能异常高效地达成共识确认(Finalize)。

DePHY消息网络的节点运行是Permissionless的，任何人都可以质押资产并运行DePHY网络节点。节点越多，网络的安全性越高，可访问性就越强。DePHY节点可以通过zk条件付款(Zero-Knowledge Contingent Payments)的方式，获得奖励。也就是说，有数据索引需求的应用，在向DePHY中继节点请求数据时，根据可否检索到数据的ZK证明，来决定向中继节点支付多少费用。

同时，任何人都可以接入DePHY网络来监听、读取数据。项目方运营的节点可以设置过滤规则，只储存与自己项目相关的DePIN设备数据。由于沉淀了原始数据，DePHY消息网络可以作为后续其他任务的数据可用层。

DePHY协议会要求中继节点在运行时至少把接收到的数据在本地存储一段时间，再把冷数据要转存到 Arweave 这种永久性的存储平台上。如果全部数据都当作热数据去处理，最终会抬高节点的存储成本，进而提高全节点运行门槛，使得普通人难以运行全节点。

通过冷热数据分类处理的设计，DePHY能大大降低消息网络中全节点的运行成本，更能应对海量的物联网数据。



### peaq

前面的两种方案都是把数据的收集存储放在链下去执行，然后Rollup到区块链上去。这是因为物联网应用产生的数据量本身极大，同时又有通信延时的要求。如果在区块链上去直接去执行DePIN交易，数据处理能力有限而且存储成本很高。

单是等待节点共识就带来不可忍耐的延时问题。peaq却另辟蹊径，自己搭建了一条公链，直接承载和执行这些计算和交易。它是基于Substrate开发的，当主网实际上线后，承载的DePIN设备增多，将会因为peaq的性能瓶颈，最终无法承载那么大量的计算和交易请求。

由于peaq没有可信任固件的功能，基本无法有效验证数据的可信。在数据存储方面，peaq直接在开发文档中介绍了如何给基于substrate的区块链接入IPFS分布式存储。

### 将数据分发到不同应用

DePIN工作流程中的第三阶段，是根据区块链应用的需求，将数据可用层的数据抽取出来，通过执行运算或零知识证明，把执行结果高效地同步到区块链上。

### 方案介绍

### IoTeX

W3bStream把这一阶段称为数据证明聚合(Data Proof Aggregation)。这部分网络由许多聚合器节点(Aggregator Nodes)组成一个计算资源池(computing resource pool), 给所有的DePIN项目共享调用。

每个聚合器节点会在区块链上记录自己的工作状态，是忙碌还是空闲。当有DePIN项目的计算需求过来时，根据链上的状态监控(monitor)选择空闲的聚合器节点去处理。



被选中的聚合器节点，会先从存储层检索需要的数据；然后根据DePIN项目的需求对这些数据做运算，并且生成运算结果的证明；最后把证明结果发送到区块链上给智能合约去验证。完成工作流程之后，聚合器节点重新回到空闲状态。

而聚合器节点在生成证明时，会用到分层聚合电路(layered aggregation circuit)。分层聚合电路包含四个部分：

数据压缩电路：类似于Merkle树，验证所有收集的数据都来自特定的Merkle树根。

签名批验证电路：批量验证来自设备的数据的有效性，每个数据都与一个签名相关联。

DePIN计算电路：证明DePIN设备按照特定计算逻辑正确执行了一些指令，例如在医疗健康项目中验证步数，或者在太阳能发电厂中验证产生的能量。

证明聚合电路：将所有证明聚合成一个单一的证明，以供Layer1智能合约最终验证。



数据证明聚合对于确保DePIN项目中计算的完整性和可验证性至关重要，为验证链下计算和数据处理提供了可靠和高效的方法。

IoTeX的收益环节也主要在这一阶段，用户可以通过质押IOTX代币，运行聚合器节点。越多聚合器的参与，也就能带来更多的运算处理能力，形成算力充足的计算层。

### DePHY

在数据分发层面，DePHY提供了协处理器来监听 DePHY 消息网络的 finalized 消息，进行状态迁移(State change)后，将数据打包压缩并提交到区块链。

状态迁移是用于处理消息的类智能合约的函数，由不同DePIN项目方定制，还包括zkVM或TEE的计算打包数据处理方案。这部分由DePHY团队向DePIN项目方提供项目脚手架(Scaffold)来开发和部署，具有很高的自由度。

除了DePHY提供的co-processor, DePIN项目方也可以根据项目脚手架将DA层的数据接入到其他基础设施的计算层，实现上链。



## 综合分析

尽管DePIN赛道火热，但物联网设备要大规模接入到区块链，仍存在技术上的障碍。本文从技术实现的角度，回顾分析了物联网设备从可信任产生数据，验证存储数据，通过计算产生证明和向区块链rollup数据的整个流程，从而支持将物联网设备集成到Web3应用中。如果你是DePIN赛道的创业者，也希望本文可以在方法论和技术设计上能对项目发展有所帮助。

在选择分析的三个DePIN基础设施中，peaq依然像六年前网上的评论一样，is just hype. DePHY 和 IoTeX 都选择了链下收集物联网设备数据，然后rollup到链上的工作模式，能够在低时延、保证设备数据可信的条件下，将物联网设备数据接入到区块链。

DePHY 和 IoTeX 又各有侧重，DePHY的DID包含了硬件功能trait的验证，双向数据传输等特点，DePHY消息网络更注重于去中心化的数据可用层，更多是作为低耦合的功能模块与DePIN项目结合；IoTeX的开发完整度很高，有完整的开发工作流程，更侧重于给不同事件绑定处理程序，偏向计算层。DePIN项目方可以根据实际需求，选择不同的技术方案去组合。