隐私数据管理系统 Nebul-alpha：将数据的所有权回归用户

刘芳璐[1]，王思喆[1]，姚信威[2]

[1](浙江健网数据科技有限公司，浙江 杭州 310003)

[2](浙江工业大学 计算机科学与技术学院，浙江 杭州 310023)

通讯作者：姚信威, E-mail: xwyao@zjut.edu.cn

摘 要：数据所有权的错位问题是当前数据管理系统中的一个难题。由于数据传输、市场分配和管理等原因，原本属于用户个人所有的数据却无法被用户所掌控，被动地造成了诸如隐私泄露，数据窃取，黑市数据交易等一系列数据管理问题。区块链技术不但为不同个体之间数据的安全传输提供了一种新的手段，而且还保证每一笔交易的痕迹都能永久被系统记录下来。Nebul-alpha 在区块链技术的基础上建立了链上链下相结合的授权认证体系，并通过去中心化的数据管理结构来确保数据所有者对数据的真正掌控，从而实现安全的、可追踪的私人数据共享。

关键词：数据所有权；隐私保护；链下计算；痕迹追踪

Nebul-alpha: A Data Management System to Reassign Privacy Data Ownership

LIU Fang-lu[1], WANG Si-zhe[1], YAO Xin-wei[2]

[1](Zhejiang WTHealth Co., Ltd., Hangzhou 310003, China)

[2](School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China)

Abstract: Ownership attribution is one of the key missing features in current data management systems. Misplacement of ownership away from individual users can lead to a series of negative consequences such as privacy breach, data stealing, and black-market data trade. Blockchain technology offers an option to technically guarantee the secure data transmission between different parties, with the trails of every transaction to be permanently recorded by the system. In this paper, we build up a blockchain-powered decentralized data management protocol Nebul-alpha to assure proper data ownership, and set up an authorization-authentication system to enable a traceable way to securely share private data.

Key words: Data Ownership; Data Privacy; Off-chain Network; Audit Trail

在数据产业中，对于隐私数据的存储和处理都是很常见的。虽然这些数据被称为“个人数据”，但它们的所有权通常不真正属于个人。对于数据公司来说，在数据管理过程中，与如何保证数据安全的优先度相比，如何更有效的挖掘数据中的价值的优先度更高[1]。数据所有权的错位导致了一个非常严重的问题，作为数据的实际占有者，数据公司们不愿意也没有动机来承担与他们所享受利益相对应的责任，从而在数据管理中埋下安全隐患[2, 3]。例如，Facebook 在 2018 年 3 月所被披露的 Cambridge Analytica 数据丑闻就是由于用户对于自己的数据缺乏掌控，用户的数据在未经用户授权的情况下被暴露给第三方[4]。用户对 Facebook 有信任并不等价于用户对于所有 Facebook 的合作公司都具有同等信任。但是，现有的数据管理系统无法让数据公司一方面产生与管理数据，另一方面却不实际拥有数据[5]。

区块链技术能够保证数据的可信性[6, 7]，同时其去中心化的运作机制使得管理由所有节点共识参与，让个人用户参与数据管理，掌控数据所有权成为了可能[8, 9]。但是，区块链技术在隐私数据管理中面临两个问题：由于信息传递的不可逆性，隐私数据在链上共享后就难以进行控制和限制其使用范围[7, 10]；以及区块链的存储容量并不足以支持拥有庞大数据量的隐私数据，隐私数据上链会对区块链造成难以负荷的拥堵[6, 9, 11]。许多

区块链技术开发者在将区块链技术应用到隐私数据管理时，都意识到了这两个问题，比如 Bluzelle 和 Iconomi，但它们提供的解决方案都不完整，有着诸多缺陷[12, 13]。Bluzelle 主要是以区块链公链为中介，将链下的数据存储方（off-chain privacy component）与基于公链的应用提供方联系起来。但 Bluzelle 在设计中，并没有提供针对数据存储方的防护处理，这使得链下的数据存储与链上的服务之间基本相互割离，链下的存储方对于所存储数据由较大的使用权限；在 Iconomi 的解决方案中，则完全没有涉及链下存储，仅仅是为数据库提供隐私保护层的服务。这样的设计，一方面让它的系统在不同数据类型与应用场合中的可调试性较低[14]；另一方面则和 Bluzelle 一样，存在着链下的存储方绕过区块链隐私保护层，而直接实现数据传输的隐患。

为了从根本上防止数据批量泄露，并解决数据所有权模糊的问题，本文设计了碎片重构算法（Fragments Reconstruction Algorithm），并基于该算法搭建了新型的数据管理框架系统 Nebul-alpha。该系统利用区块链技术的优势，以让所有参与方共识的方式来进行数据的管理，消除了后台操作的可能性。同时，数据完全由数据所有者控制，只有数据所有者才能向他人授权使用数据，并且所有数据访问都会在区块链上留下可追踪的，公开的，不可消除的痕迹（audit trail），从而实现安全的、可追踪的私人数据共享。

本文第一节对于所设计的 Nebul-alpha 系统的架构进行介绍，包括所实现的功能目标，框架流程，以及数据结构；第二节对于 Nebul-alpha 的各参与方进行了详细的解释说明，包括各参与方的权限，API，以及要求；第三节对于 Nebul-alpha 所使用的碎片重构算法进行了说明；第四节则配合 Nebul-alpha 的特点，解释说明其使用场景；最后对于 Nebul-alpha 的特性进行总结。

1 系统架构

1.1 系统要求

对于隐私数据的管理来说，让数据能够安全地被存储，并能让数据所有者完全控制数据的分享情况，是至关重要的。与此同时，该系统还需要有足够的兼容性与拓展性，以便满足对不同类型数据的管理。综上所述，本文所设想的系统必须满足以下特性:

(1) 未经数据所有者授权，任何人都无法获取到信息;

(2) 文件的获取行为都会在区块链上留下无法消除的痕迹，并且该痕迹可被公开查阅;

(3) 所有者的授权方式可以根据应用场景的不同来进行调整，并且授权具有时效性;

(4) 数据安全与数据量之间的相关性小。

第一个特性使数据所有者可以完全控制其自己的数据；当数据泄漏发生在线下时，第二个特性可以帮助调查和确定责任的归属；第三个特性可以防止数据被重入攻击（reentrancy attack）[15]，并使数据所有者对于数据的分享行为限定在当下，而没有授予未来更新的数据的权限；最后一个特性则使得大规模的批量数据泄露几乎不可能发生。在具体设计过程中，为了满足前面提到的四点系统特性，可以采用如下方法：

方法 1：数据存储在无意义的碎片中，只在阅读时才能重建为有意义的内容。该方法的核心是碎片重构算法，这同时也是在授权时形成痕迹的关键设计。一段私有数据被分段并存储在不同的存储节点中。这些数据仅被重新整合在一起时，才会有意义。而只有当所有相关的存储节点对于授权产生共识并通过验证后，数据读取者才能获得所有的所需碎片，从而进一步拼凑出完整文档。数据读取者无法用只向一个存储节点发起请求的方式，来通过后门获取数据。出于同样的原因，由于痕迹基于共识被写在区块链上，单个存储节点不可能帮助数据读取者在不留痕迹的情况下秘密读取数据。

方法 2：所有数据都在系统中被加密。该方法保证数据内容不会泄露给没有加密密钥的人，这是保护数据安全的基本方法，不过并非所有存储系统都可以提供此功能。但是，只有加密是不够的，因为加密密钥一旦交给数据阅读者后，数据就将开始流通。因此，使用方法 1 碎片重构算法而搭建的授权层是十分必须的。

方法 3：用于加密数据的密钥受数据所有者的私钥保护。该方法确保数据泄漏相互间的不相关性，因为私钥只被数据所有者持有，即便某一个数据所有者的数据被泄露，也不会波及到其他的数据所有者所持有的

数据。

方法 4：数据处理机制灵活，可以根据应用场景提供不同级别的访问控制。该方法对于支持各种数据的应用很重要，例如，财务数据，健康数据，以及其他数据市场。不同的使用场景对数据访问控制的细化程度有不同的要求。值得注意的是，该系统无法阻止数据读取者在获取文档后，通过线下的方式将之泄露。因此，数据所有者应该只向可信任的第三方授予权限。

1.2 系统框架流程

综合以上四种方法后，Nebul-alpha 系统的框架流程图可以展示如图 1 所示。系统设计一共涵盖了 3 个层，即客户端（client），链下存储层（off-chain storage nodes），以及共识层（blockchain consensus layer）。实现的功能包含三部分，即写入授权（write auth），写入信息（write）与读取信息（read）。其中，写入授权根据系统应用的具体场合与不同的数据类型，可视为可选步骤。在实际处理数据过程中，write 客户端负责数据的加密与碎片化，并将数据写入提交共识层进行验证与记录后，在链下存储层进行写入存储；链下存储层在将数据碎片写入存储后，将提交 metadata 到共识层记录；read 客户端在读取数据时，提交读取请求到共识层来进行验证和记录；通过共识验证的请求将反馈到存储层来，并由存储层发送数据包到 read 客户端；read 客户端在获得所有数据碎片后，再进行重组与解密。让隐私数据的处理与存储都在链下进行，而区块链共识层则作为控制器，依靠数字签名和可编程权限来实现访问控制与行为记录。

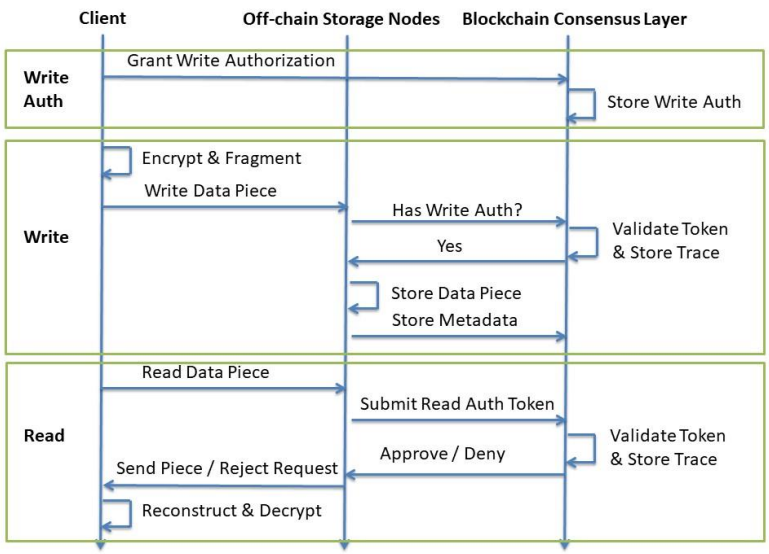

Fig.1 Overview of Nebul-alpha data management system

图 1　Nebul-alpha 系统架构流程说明图

1.3 数据结构

Nebul-alpha 系统采用扁平命名方法命名文件。每个用户将被分配一个 id - UID。 UID 可以是政府颁发的 ID，也可以是 PKI 密钥对的公钥[16]，这取决于具体的用途。例如，如果数据是用户的信用记录或医疗档案，则第一种方式较好。如果系统用于存储一些 ad-hoc 数据项，则应采用第二种方式。每个 UID 与都会有与之相关联的密钥对，密钥对将用于在系统中创建签名或加密数据；在这种情况下，用户可以被描述为以下数据结构：

User {

UID (can be the same as PublicKey)

PublicKey

PrivateKey

}

在 Nebul-alpha 系统中，每个数据项或文件都与所有者相关联，所有者可以决定是否将信息透露给第三方。所有者可以拥有多个数据项或文档，每个文档都有一个 ID，称之为 DID。 DID 由两部分组成，DID =（owner’s UID, d），其中 d 是所有者库中特定文档的 ID。不同的所有者可以拥有相同 d 的文档。通过其所有者的 UID 和 d 的组合来处理文档：

Document {

UID

d

content

}

本系统将文档拆分为碎片，并将不同碎片存储在不同的数据存储节点中。这是每次读取生成痕迹的关键：只有当存储文档的所有数据管理员就数据所有者授权读取的有效性达成共识时，读取者才能获得完整的文档。当所有数据管理员达成共识时，他们将在区块链上写入读取信息并将数据块发送给读者。每个数据块还应具有 ID，CID，CID = DID + c。其中，当把一份文档拆分成 C 份时，c 则是 0 到 C-1 之间的数字。 C 是可配置的。

上述基本的数据结构，还可以根据具体的需求来进行扩充。有时读取者可能需要阅读某一类文档而不是单个文档。例如，银行需要一个人的完整信用记录才能批准抵押贷款。信用记录包含可以来自多家银行的所有信用文件。在这种情况下，标签的概念在这里就很有用。除了 CID 和所存储的内容之外，数据存储者还为每个文档存储标签，以便当读取者通过标签请求时，他们还可以返回相关文档。

2 系统参与方

本系统一共涉及到四类参与者：数据编撰者，数据所有者，数据读取者和数据存储者。

2.1 数据编撰者（data writer）

数据编撰者是写入数据的主体。数据编撰者可以是数据所有者，也可以是已被数据所有者授予访问权限的第三方。例如，当患者上传他/她的信息时，编撰者就是所有者。当医生提交就诊记录时，编撰者则与所有者是不同个体，编撰者代表所有者写入数据。“代表”包含两层含义：首先，编撰者上传有关所有者的信息，例如医疗保健信息或信用记录；其次，写入的数据还应归属于所有者拥有，只有所有者才能决定是否向其他方公开数据。授权的方式是将编撰者持有公钥，用所有者的私钥进行签名。当编撰者尝试将一条信息放入系统时，所有节点都将验证所有者是否授权写入者操作。原则上来说，所有者应该只授权受信任的编撰者；但实际上，可能仍然有恶意编撰者输入有关对所有者不利的负面数据或垃圾信息。因此，根据使用痕迹来设立的评级系统是十分必要的，即根据编撰者的来源，数据类型的历史反馈情况，来对写入的信息区分不同的可靠性级别。

在为 UID 写入 DID 时，数据编撰者的工作流程是这样的：

(1) 生成加密算法 E，并且使用 E，将文档加密为 E(document);

(2) 根据 erasure coding 将 E(document)划分为 C 个碎片，并将 C 个碎片分发到 C 个存储节点。每个碎片都附有编撰者的公钥，签名，所有者的公钥，以及 CID;

(3) （可选）存储节点只有验证通过所有者对于编撰者的授权后，才会接受并存储文档碎片；

(4) 编撰者使用所有者的公钥和所有数据块的 CID 将加密算法 E 本身再进行加密，并将加密后的 E 存储到区块链上。此时，节点也将对所有者对编撰者的授权的有效性进行验证。加密算法 E 由所有

者的公钥加密并存储在链中，以便所有者可以从区块链中提取该加密算法并将其用于以后的授权。

```
Write(/* Writer’s public key */ w_pk, /* owner’s public key */ o_pk, content) {
   d = RandGenId();
   E = GenerateEncryptionKey();
   EncryptedContent = E(content);
   Pieces[] = SplitToPieces(EncryptedContent, C);
   for i in {0.. C - 1} {
      Signature = Sign(Hash(Pieces[i]), o_pk, d, i, w_pk);
      SendToStorageNode(Pieces[i], o_pk, d, i, w_pk, Signature);
   }
   WriteOnChain(o_pk(E), o_pk, d);
}
```

2.2 数据所有者（data owner）

数据所有者是拥有数据的主体。通过拥有数据，所有者可以向一方（包括所有者本身）提供写入和读取的授权。如没有所有者的写入授权，他人无法写入有关所有者的信息。例如，医生或医院需要患者授权，才能输入患者的医疗信息。这是为了防止有恶意信息编写者随意编写不准确的信息。但是，写入授权是可选的，具体取决于用途。另一方面，没有读取授权，他人无法读取所有者拥有的信息。这可以确保没有明确授权的情况下，任何人都无法读取用户的私人数据。数据所有者在链下向接收数据读取者发出的“授权”请求，即读取者将 DID =（UID, d）和及其签名一起发送给所有者。 如果所有者想要允许读取者读取数据，所有者将使用其所有者私钥对消息进行签名，并发送 authorization token 的签名消息给读取者。与此同时，所有者还需要通过安全的通道将用于加密文档的密钥 E 发送给读取者。

请注意，authorization token 可以具有附加功能，例如，只能使用有限次数。 由于 authorization token 的使用将被记录在区块链上，因此可以轻松验证其使用的次数。 利用 authorization token 的过期机制，读取者将对所有者的数据进行有限的访问，即只有当下的信息，而不是所有后续的更新信息，从而增强了安全性。

```
// This is an offchain method
AuthorizeRead(r_sig(DID), r_pk) {
    UID, d = Decode(r_sig(DID), r_pk);
    if(!IsMyID(UID)) return error;
    Token = Sign(r_pk, DID);
    Send({to: reader}, Token, r_pk(E));
}
```

2.3 数据读取者（data reader）

数据读取者是想要读取数据的主体。只有在数据所有者授予权限的情况下，数据读取者才能读取某个数据。读取者会将 authorization token 发送给存储数据的区块链节点。 区块链节点将验证所有者和读取者的签名，并且存储对应 DID 的数据块的存储节点将把相应的数据块发送到读取者。 同时，这些存储节点将在区块链上添加带有时间戳的阅读请求，从而形成痕迹。

```
Read(o_pk, DID) {
   Token, E = RequestAuthorization(o_pk, DID);
   Pieces[] = RequestRead(o_pk, DID, Token);
   EncryptedFile = RestoreFile(Pieces);
   File = E(EncryptedFile)
}
```

2.4 数据存储者（data keeper）

数据存储者是存储数据的节点，负责运行共识算法和生成区块。数据存储者不仅基于其提供的存储服务

的数量，质量和持续时间，而且还根据其接收的读取请求的数量来取得奖励。读取者将其阅读费用支付给所有者（作为披露其数据的奖励）和数据存储者（作为存储费用和区块链服务费）。

```
ServeReadRequest(o_pk, DID, Token, r_pk) {
    TokenValid = ValidateToken(o_pk, DID, Token, r_pk);
    Piece = FindPiece(DID);
    SendPiece({to: reader}, Piece);
}
```

以个人健康数据和财务数据为例，其中所包含的信息都是十分敏感且庞大的。它们必须被存放在链外，否则整个区块链会变得非常笨重。同时，由于这些数据价值很高，所以它们的存储必须非常耐用可靠。出于对数据的敏感度性，系统所需要的可靠性和存储的大容量等因素的考量，让任意的节点都能作为数据存储者并不理想。因此，在 Nebul-alpha 系统中将只允许信誉良好的机构提供存储节点，而其中的耐久性将由经济激励措施来保证。请注意，由多个参与方形成的区块链不同于中心化的系统，其中的权限由多方组成，形成共同监督方。即使是由超级存储节点运行，与单个公司运行的传统存储系统相比，Nebul-alpha 系统仍然具有开放性，因为所有决策都是由系统中的所有节点做出的，而不是由系统中能够打开后台与未经授权的第三方共享数据的中心化的系统做出的。

存储节点必须满足以下标准：首先且最重要的是，它们应该能够长期持有数据，并且有极小的破坏或丢失数据的可能；其次，存储节点应该能够在大多数时间保持在线状态。在收到请求时，当某个节点离线时，离线的节点和丢失数据的节点，反应完全没有区别。虽然从技术上讲，在一定程度的存储节点丢失数据或离线的情况下，本系统仍然可以通过 erasure coding algorithm 来恢复完整文档[17]，但符合标准的存储节点将可以大大提高系统计算效率，降低运行成本，故这两个标准仍然是存储节点证明其值得信赖的必要条件。

3 碎片重构算法

确保本系统所需要的严格的读取授权策略和完整访问痕迹的，是碎片重构算法。该算法首先对文件进行加密，然后将其拆分为多个部分。密钥本身是文件的第一级保护。用户必须通过安全的网络通道从数据所有者处获取解密密钥才能读取该文件。但是，碎片重构算法提供的不仅仅是简单的加密文件存储系统。每个文件片段将存储在不同的数据存储节点中。当读取者请求读取文件时，每个数据存储节点将验证读取者是否具有适当的授权。如果区块链网络就授权验证达成共识，数据管理员仅将自己存储的片段发送给读取者。同时，读取信息将被记录在新的区块中，就像其他区块链系统中的交易一样。这种设计的目的是确保即使读取者在窃取解密密钥的情况下，也无法仅从文件的一部分中获取任何有意义的信息。单一的碎片无法被解密将通过 all-or-nothing transformation 来实现[18, 19]。因此，这就有了只有本系统提供的第二层保护。只要有足够数量的可信的数据存储节点，即便存在一小部分虚假或有故障的节点会在不验证授权的情况下向读取者发送一些部分，读取者仍然无法不通过共识层验证获得文件的所有部分。同时，只要有足够数量的可信节点，每个读取请求（被拒绝或接受）都将被记录在区块链上，生成成不可擦除的公开的访问记录。

3.1 算法可行性

一个自然要问的问题是系统可以维持多少不可靠的存储节点。不可靠的数据管理器节点可以造成存储在其上的数据的丢失，因此本系统需要保证所有数据都由可靠的恢复机制正确管理，以便始终可以重建完整的文件。本系统使用 Reed-Solomon 或其嵌套版本的 erasure coding 将文件剪切成碎片[20, 21]，这样即使丢失一些碎片，文件仍然可以重建。这两种版本都提供了各种选项来在存储成本和数据丢失阻力之间进行权衡。

3.2 奖励和支付

存储节点根据它们提供的存储量以及这些存储的使用时间来获得奖励。生成新区块时会定期支付奖励。根据使用场景，数据编撰者，数据读取者或数据所有者可能需要为系统的运行支付费用。例如，在医院 - 患

者情景中，如果研究机构对数据感兴趣，则患者可以授权机构读取数据；此时，研究机构应该支付费用给患者和医院。在另一个用例中，用户可以将他/她的临时数据放入系统中；在这种情况下，同时作为数据所有者和数据编撰者的用户，应该为存储付费。

3.3 监控机制

碎片重构算法也提供技术监控机制，以帮助检测那些行为不端/不可靠的存储节点。每个诚实节点可以定期向存储节点发出 challenge message 以请求获取某些文件碎片。我们将例如 checksums 的文件碎片的 metadata 存储在区块链上，从而发送 challenge message 的节点可以在不读取文件的情况下验证加密的内容。如果存储节点无法对 challenge 做出回应，则会对其进行更仔细检查以确保它是否仍然持有应存储的数据。

4 系统应用

本数据管理系统可以被普遍使用，并应用于各种数据系统中。下面列举三个可以应用 Nebul-alpha 系统的使用场景。

首先，Nebul-alpha 可以被应用到医疗系统中。患者自己可以在系统中上传他们的健康信息，或者他们可以授权他们的医生将他们的医疗记录和处方放入系统中。来自不同来源的医疗健康信息将被标记为不同的可靠性级别，其中由专业医疗服务提供者所提供的信息将会有更高级别的可靠性。想要查看医疗信息的第三方，例如科研机构或其他医生，必须先请求患者授权来取得读取权限。只有当患者愿意提供数据时，其他机构才能阅读数据。

第二个使用场景是信用信息。在这种情况下，银行，公用设施提供商以及向用户发送账单的任何公司都可以写下有关客户信用记录的信息。此时，这些机构是数据的编撰者；虽然用户是数据所有者，但他们无法编辑和修改他们自己的信用信息。用户仍然有权对第三方设定读取权限，例如当他们想要申请银行信用卡，或者在需要向零售商申请分期付款时。这些第三方可以通过了解客户自己标记的最低的可靠性标准信用来实现读取信息。不同来源的信用信息将会有不同的可靠性。

第三个使用场景则是数据市场。目前，许多数据的所有者拥有具有价值的数据但无法将它们转化为收益，因为没有系统可以安全且可追踪地分享这些潜在的私密数据。使用本系统，数据所有者可以授权有兴趣使用这些数据的用户前来进行精细可控的读取访问。访问者将在区块链上留下的不可擦除 audit trail 的则可用于计费。

5 总结

Nebul-alpha 系统设计的根本目的是解决隐私数据的所有权确权问题，即将数据的所有权回归用户。通过解决这个问题，Nebul-alpha 为数据的存储于分享提供了更为安全的渠道。相对于提供加密后的传统存储系统，Nebul-alpha 具有以下三点优势：第一，对所有数据的读取都会记录在不可擦除的，基于共识过程而自然形成的历史痕迹将被记录在区块链上，这个属性在多个实际使用案例中至关重要。例如，在数据市场中，完整的读取历史可以作为计费的证据。第二，在 Nebul-alpha 系统中，授权将由所有存储节点来进行验证，因此所谓的后门将难以或不可能完成。第三，加密密钥的泄露并不是无法弥补，因为 Nebul-alpha 提供了对数据隐私的第二层保护。

References:

[1] Chen CP, Zhang CY. Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. Information Sciences. 2014 Aug 10;275:314-47.

[2] Spiekermann S, Acquisti A, Böhme R, Hui KL. The challenges of personal data markets and privacy. Electronic Markets. 2015 Jun 1;25(2):161-7.

[3] Kish LJ, Topol EJ. Unpatients—why patients should own their medical data. Nature biotechnology. 2015 Sep 8;33(9):921.

[4] Granville K. Facebook and Cambridge Analytica: What You Need to Know as Fallout Widens. The New York Times. 2018.

[5] Jagadish HV, Gehrke J, Labrinidis A, Papakonstantinou Y, Patel JM, Ramakrishnan R, Shahabi C. Big data and its technical challenges. Communications of the ACM. 2014 Jul 1;57(7):86-94.

[6] Pilkington M. 11 Blockchain technology: principles and applications. Research handbook on digital transformations. 2016 Sep 30:225.

[7] Crosby M, Pattanayak P, Verma S, Kalyanaraman V. Blockchain technology: Beyond bitcoin. Applied Innovation. 2016 Jun;2:6-10.

[8] Yue X, Wang H, Jin D, Li M, Jiang W. Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control. Journal of medical systems. 2016 Oct 1;40(10):218.

[9] Zyskind G, Nathan O. Decentralizing privacy: Using blockchain to protect personal data. In Security and Privacy Workshops (SPW), 2015 IEEE 2015 May 21 (pp. 180-184). IEEE.

[10] Angraal S, Krumholz HM, Schulz WL. Blockchain technology: applications in health care. Circulation: Cardiovascular Quality and Outcomes. 2017 Sep;10(9):e003800.

[11] Swan M. Blockchain thinking: The brain as a decentralized autonomous corporation [commentary]. IEEE Technology and Society Magazine. 2015 Dec;34(4):41-52.

[12] Bains P, Murarka N. A Decentralized Database for The Future. Bluzelle PlaTorm Pte. Ltd. 2017 Dec 11.

[13] Zagar T, Valjavec J, Batagelj Z, Kovac E, Lekse A. Open Fund Management Platform to Disrupt the Investment Industry. ICONOMI Inc. 2016 Jul.

[14] Spanos N, Martin AR, Dixon ET, Geros AS, inventors; Blockchain Technologies Corp, assignee. System and method for creating a multi-branched blockchain with configurable protocol rules. United States patent US 9,608,829. 2017 Mar 28.

[15] Luu L, Chu DH, Olickel H, Saxena P, Hobor A. Making smart contracts smarter. InProceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security 2016 Oct 24 (pp. 254-269). ACM.

[16] Weise J. Public key infrastructure overview. Sun BluePrints OnLine, August. 2001 Aug:1-27.

[17] Baker D, Carpentier PR, Klager A, Pierce A, Jonathan RI, Turpin R, Yoakley D, inventors; Caringo Inc, assignee. Erasure coding and replication in storage clusters. United States patent US 8,799,746. 2014 Aug 5.

[18] Rivest RL. All-or-nothing encryption and the package transform. InInternational Workshop on Fast Software Encryption 1997 Jan 20 (pp. 210-218). Springer, Berlin, Heidelberg.

[19] Resch JK, Leggette W, inventors; International Business Machines Corp, assignee. Utilizing a deterministic all or nothing transformation in a dispersed storage network. United States patent US 9,231,768. 2016 Jan 5.

[20] Reed IS, Solomon G. Polynomial codes over certain finite fields. Journal of the society for industrial and applied mathematics. 1960 Jun;8(2):300-4.

[21] Blaum M, Hafner JL, Hetzler SR, inventors; International Business Machines Corp, assignee. Nested multiple erasure correcting codes for storage arrays. United States patent US 8,433,979. 2013 Apr 30.