软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, [doi: 10.13328/j.cnki.jos.000000]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

典型匿名隐私保护方案比较*

白 健[1], 王 震[1], 陈宇翔[1], 范 佳[1], 安红章[1]

(保密通信重点实验室,四川 成都 610041)

通讯作者: 白健, E-mail: baijian_30@163.com

摘 要: 区块链是随着比特币等数字加密货币的日益普及而兴起的一门全新技术，它提供了一种去中心化的信任模型，引起了金融行业、科研机构及政府部门的高度重视。区块链账簿的公开性决定了账簿必须进行匿名隐私保护，从而防止恶意攻击者分析账簿数据，窃取个人隐私。本文通过对国际典型的匿名隐私保护算法进行分析介绍，主要包括 CryptoNote 匿名隐私保护技术、Monero 匿名隐私保护技术、Fabric 匿名隐私保护技术、IDMixer 匿名隐私保护技术、Zcash 匿名隐私保护技术，对方案的安全性和效率进行比较分析，为区块链匿名隐私保护技术的发展和应用推广提供技术基础。

关键词: 区块链; 匿名隐私保护; CryptNote; Monero; Fabric; IDMixer; Zcash

中图法分类号: TP309

中文引用格式: 白健,王震,陈宇翔,范佳,安红章.典型匿名隐私保护方案比较.软件学报. http://www.jos.org.cn/1000-9825/0000.htm

英文引用格式: Bai J, Wang Z, Chen YX, Fan J, An HZ. Comparison of Typical Anonymous Privacy Schemes. Ruan Jian Xue Bao/Journal of Software, 2016 (in Chinese). http://www.jos.org.cn/1000-9825/0000.htm

Comparison of Typical Anonymous Privacy Schemes

Bai Jian[1], Wang Zhen[1], Chen Yuxiang[1], Fan Jia[1], An Hongzhang[1]

[1](Science and Technology on Communication Security Laboratory, Chengdu 610041, China)

Abstract: Blockchain is a new technology that emerges with the increasing popularity of crypto-currencies such as bitcoin. It provides a decentralized trust model that has attracted the attention of the financial industry, research institutions and government agencies. The transparency of blockchain determines that the ledger must be protected by anonymous privacy, thereby preventing malicious attackers from analyzing ledger data and stealing personal privacy. This paper analyzes the typical international anonymous privacy protection

* 基金项目: 国家重点研发计划（2016YFC0801004）;
Foundation item: National Key R&D Program of China (2016YFC0801004);
收稿时间: 0000-00-00; 修改时间: 0000-00-00; 采用时间: 0000-00-00; jos 在线出版时间: 0000-00-00
CNKI 在线出版时间: 0000-00-00

algorithms, including CryptoNote anonymous privacy protection technology, Fabric anonymous privacy protection technology, IDMixer anonymous privacy protection technology, and Zcash anonymous privacy protection technology. The security and efficiency of the schemes are compared and analyzed. These technologies provide a technical basis for the development and application of blockchain anonymous privacy protection technology.

Key words: blockchain; privacy; anonymous; CryptoNote; Fabrics; Monero; Fabric; IDMixer; Zcash

2008 年，中本聪首次提出了区块链的概念[1]，它是近年来最具革命性的新兴技术之一。区块链技术发源于比特币，其以去中心化方式建立信任等突出特点，对金融等诸多行业来说极具颠覆性，具有非常广阔的应用前景，受到各国政府、金融机构、科技企业的高度关注。2015 年，世界许多重大组织，包括高盛、花旗银行、美国央行、英国央行等机构纷纷在区块链上面进行投资。华尔街日报宣称区块链是最近 500 年以来在金融领域最重要的突破。2016 年起，区块链在中国受到了极大重视[2]。中国人民银行宣布要使用数字货币，许多国内的组织机构开始进行区块链的投资与研究。2017 年，国家各地方政府纷纷出台政策，鼓励和支持区块链技术的发展。2018 年，习近平主席在在中国科学院第十九次院士大会、中国工程院第十四次院士大会中提到“以人工智能、量子信息、移动通信、物联网、区块链为代表的新一代信息技术加速突破应用”[3]。

区块链是在点对点(P2P)网络上构建的分布式数据库系统。利用非对称加密算法进行签名打包的每个数据存储单元被称为区块，区块与区块通过杂凑算法相连形成的链条称为区块链。区块链具有如下的特点[4]：

（1） 去中心化。区块链数据的存储、传输、验证等过程均基于分布式的系统结构，不依赖于一个中心化的硬件或管理机构，少量节点的损坏或失去都不会影响整个系统的运作。

（2） 可靠数据库。区块链系统的数据库采用分布式存储，通过共识算法实现各个系统之间区块链的账簿，除非能控制系统中大部分共识权限，否则在节点上对数据库的修改都将是无效的。

（3） 开源可编程。区块链系统中的智能合约层可以通过根据应用规则进行编程，从而实现将各种应该系统对接到区块链系统中。

（4） 安全可信。区块链技术采用非对称密码学算法对交易进行签名，使得交易不能被伪造；同时利用哈希算法保证交易数据不能被轻易篡改，最后借助分布式系统各节点的工作量证明等共识算法形成强大的算力来抵御破坏者的攻击，保证区块链中的区块以及区块内的交易数据不可篡改和不可伪造，具有极高的安全性。

这些特点决定了其在互联网领域的广泛用途，主要包括：互联网金融投资管理平台[5]、资产登记管理平台[6]、保险登记和管理[7]、游戏虚拟资产管理[8]等等。

然而，基于区块链设计安全协议需要考虑区块链的公开验证和隐私保护之间的矛盾，基于区块链的安全协议一般是通过交易来完成协议的安全目标。而交易是公开的，区块链中每一个节点都可以读取交易数据，验证交易数据是否正确。如果交易数据中涉及隐私的内容，矛盾就产生了。因此，区块链上的隐私保护是一个热点研究问题。

区块链最广泛的应用领域之一是电子货币，隐私性是电子货币最重要的安全因素。T. Okamoto 和 K. Ohta 描述了理想的电子现金应该满足的六个标准，其中对隐私的描述是用户和其购买行为之间的关系对任何人都是不可追溯的[9]。交易的隐私性一般要满足下面两条性质：

（1）匿名性：对于每个交易，不能确定交易的发送者或接收者。

（2）不可链接性：对于任何两个输出交易，不可能证明他们被发送给同一个人。

（3）交易数据隐私性：对于交易内容，智能运行交易相关方获取。

然而，原始的比特币系统并不完全满足隐私性。人们常说的比特币的匿名性是指用户通过地址来使用比特币，账本中的交易只记录了地址和比特币数量，从交易中无法直接得到用户的真实信息。然而，比特币地址不具备真正的匿名性，更准确的说，应是假名性，就像用户在网络论坛上使用的代号一样，虽然不知道用户是谁，但用户的一言一行都可关联到这个代号上，实现交易的可追溯。同时，比特币的每一笔交易都会公开记录在区块链账本上，任何人都可以查阅。只要通过分析每个地址发生过的交易，就可以发现很多的账号

之间的关系[10--13]。因此，比特币系统也不满足不可链接性。区块链中的隐私保护是一个急需解决的重要问题。

本文分析介绍区块链上五种典型的匿名隐私保护方案：CryptoNote 匿名隐私保护技术、Monero 匿名隐私保护技术、Fabric 匿名隐私保护技术、IDMixer 匿名隐私保护技术和 Zcash 匿名隐私保护技术，研究了五种匿名隐私保护方案的特点，并对方案的安全性、时间效率、空间效率进行了比较。

1 CryptoNote 匿名隐私保护技术

CryptoNote 是由 Saberhagen[14]提出的一种能够克服比特币中隐私保护问题的方案。

符号及术语说明：

q: 一个素数；

E_q：F_q 上的一条椭圆曲线;

G: 椭圆曲线上的一个阶为素数 l 的基点;

H_s: 密码 Hash 函数$\{0,1\}^* \rightarrow F_q$;

H_d: 确定型 Hash 函数 $E(F_q) \rightarrow E(F_q)$;

椭圆曲线私钥: $a \in [1,l-1]$, $b \in [1,l-1]$;

椭圆曲线公钥: $A=aG$, $B=bG$;

一次密钥对: (a,A)或(b,B);

用户私钥: (a,b);

用户公钥: (A,B);

监管密钥: (a,B).

在 CryptoNote 的交易中，交易发送者首先产生一个秘密随机值，并利用秘密随机值和接收者的公钥派生出一个交易地址，并将这笔交易进行广播。交易接收者利用自己的私钥验证网络中广播的交易是否属于自己，如果属于自己，恢复出对应的私钥。Alice 支付一笔款项给 Bob 的具体协议流程如下（参考图 1，图 2）。

(1) Alice 获得 Bob 的公钥(A,B);

(2) Alice 产生一个随机数 r∈[1,l-1],计算出交易公钥 P= H_s (rA)G+B 及 R=rG;

(3) Alice 将 P 作为交易输出地址，将交易信息及 R 广播到网络中;

(4) Bob 用他的私钥(a,b)检测网络中新广播的交易，计算 P'= H_s (aR)G+B. 如果 P=P'，则这笔交易是发送给 Bob 的;

(5) Bob 恢复出交易对应的私钥 x=Hs(aR)+b，以便在以后的交易中通过用 x 签名使用这笔输出.

对于交易接收者 Bob 而言，若发送者每次使用的随机数 r 不同，则交易的目标地址 P 是不同的。所以，对于 Bob 的所有接收交易，若随机数 r 不同，则交易是不可关联的。对于交易发起者 Alice 而言，假设他的两对公私钥为(a',A')和(b',B')。当 Alice 发起交易时，使用派生的私钥 $x=H_s(a'R)+b'$对交易进行签名，公布派生公钥 $P'=H_s(rA')G+B'$及 $R=rG$。若 Alice 每次交易使用不同的随机数 r，则每次交易发起使用的签名密钥是不同的，从而实现交易的匿名性。

此外，CryptoNote 方案还支持交易的审计功能。若审计者 Carol 已知接收方 Bob 的审计密钥(a,B)，则他可以验证网络中的交易是否是发送给 Bob 的。同理，若审计者 Carol 已知发送方 Alice 的审计密钥(a',B')，也可以验证网络中的交易是否是 Alice 发起的。在原始的比特币系统中，并不支持交易的可审计。

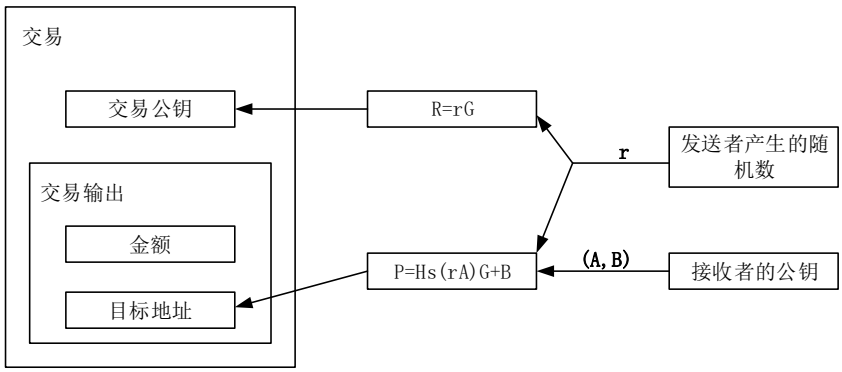

Fig.1 Transaction structure of CryptoNote

图 1 CryptoNote 交易结构

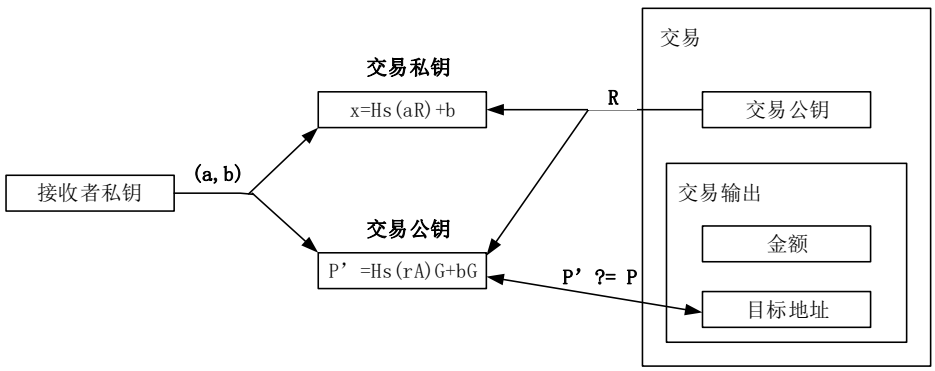

Fig.2 Transaction verification of CryptoNote

图 2 CryptoNote 中的交易收入验证

同时，CryptoNote 方案中还使用环签名技术实现匿名隐私保护，将会在下一节中进行分析介绍。

2 Monero 匿名隐私保护技术

Monero 是创建于 2014 年的一种开源加密货币[15]，其主打的便是对于用户的匿名隐私保护，而使用的核心方案便是环签名技术[16]。环签名匿名隐私保护技术的核心思想是用户在进行签名或者验签时将自己的签名公私钥同其他用户的签名公私钥一起构成密钥集合进行使用，从而保证用户交易的不可链接性。

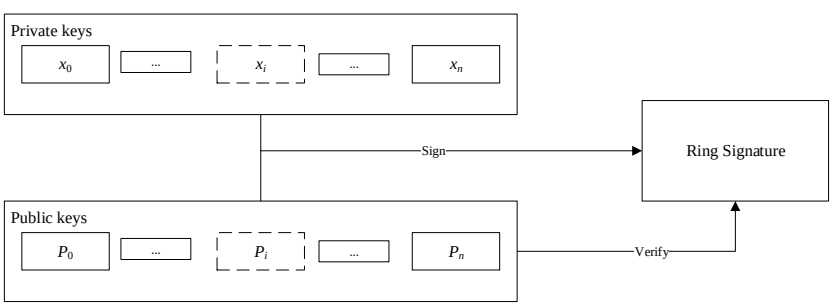

图 3 环签名匿名隐私保护技术

环签名算法核心包括四个环节：

(1) 密钥生成环节，选取公共参数，生成公私钥对(P,x)和公钥 I；

(2) 签名阶段，选取消息和签名公钥集合 S_1，输出签名结果 σ 和公钥集合 $S=S_1\cup\{P\}$；

(3) 验证阶段，选取消息 m 和公钥集合 S，签名 σ，输出 true 或者 false;

(4) 确认阶段，使用集合 $I=\{I_i\}$，和签名 σ，输出 linked 或者 indep。

Monero 中使用的详细环签名算法如下：

(1) 密钥生成，选取随机私钥 x，计算公钥 $P=[x]G$，同时计算密钥镜像 $I=[x]H_p(P)$；

(2) 签名

① 选取其他用户的公钥生成公钥集合 $S_1=\{P_i\}$，$i\in\{1,n\}$，同时使用公钥 P 和密钥镜像 I，其中 P 的公钥序号为 s，记 P 为 P_s，组成新的公钥集合 $S=S\cup\{P_s,I\}$;

② 选取 n 个随机数$\{q_i|i=0,\ldots,n\}$和$\{w_i|i=0,\ldots,n\}$，计算

$$L_i=\begin{cases}[q_i]G, i=s\\ [q_i]G+[w_i]P_i, i\neq s\end{cases}\text{和}\ R_i=\begin{cases}[q_i]H_p(P_i), i=s\\ [q_i]H_p(P_i)+[w_i]I, i\neq s\end{cases}$$

③ 计算非交互式承诺[17]，$c=H_s(m,L_1,\ldots,L_n,R_1,\ldots,R_n)$；

④ 计算签名值 $\sigma=(I,c_1,\ldots,c_n,r_1,\ldots,r_n)$

$$c_i=\begin{cases}w_i, i\neq s\\ c-\sum_0^n c_i, i=s\end{cases}\text{和}\ r_i=\begin{cases}q_i, i\neq s\\ q_s-c_sx, i=s\end{cases}$$

(3) 验签

① 计算$\begin{cases}L_i'=[r_i]G+[c_i]P\\ R_i'=[r_i]H_p(P_i)+[c_i]I\end{cases}$;

② 验证$\sum_0^n c_i \overset{?}{=} H_s(m,L_0',\ldots,L_n',\ldots,R_0',\ldots,R_n')$，相等通过，否则错误。

(4) 确认，验证者使用存储的 $I=\{I_i\}$，确认 I 是否被使用过。

3 Fabric 匿名隐私保护技术

为了克服比特币交易效率低及交易确定性等问题，Linux 基金于 2015 年 12 月启动了名为“超级账本”的开源项目，旨在推动各方协作，共同打造基于区块链的企业级分布式账本底层技术，用于构建支撑业务的行业应用和平台。Fabric 是超级账本进入孵化状态 5 个项目之一。Fabric[18]项目的目标是实现一个通用的权限区块链的底层基础框架，它克服了比特币等公有链项目的吞吐量低、共识算法低效、无隐私等缺陷，使得用户能够方便地开发商用应用。

Fabric 利用两类证书来实现交易的隐私性与可审计性。这两类证书分别是注册证书 Ecerts 和交易证书 Tcerts，其中 Ecerts 代表用户的身份。用户在 Fabric 系统中进行注册时，获得由注册证书颁发机构（ECA）颁发的注册证书 Ecert。当用户需要交易时，从交易证书颁发机构（TCA）获得一批由 Ecert 派生出的交易证书 Tcerts。Fabric 的层级公共密钥技术设施结构可参考图 3。

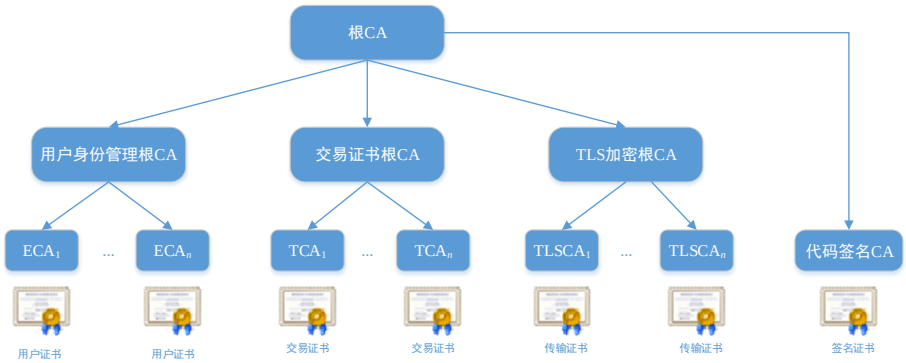

Fig.4 Fabric's Hierarchical Public Key Infrastructure

图 4 Fabric 的分层公共密钥基础设施

交易证书 Tcerts 包含如下信息：

（1） *TcertID*：Tcert 证书的标识符；

（2） *TcertIndex* 密文：$AES\_Encrypt_{TCertOwner\_EncrypKey}(TcertIndex||常数)$，用于用户恢复出 Tcert 对应的私钥；

（3） *Signature_Verification_TCertPub_key*：签名验证公钥；

（4） *Key_Agreement_TCertPub_key*：密钥协商公钥;

（5） 证书有效期。

其中 *TcertIndex* 是生成签名验证公钥和密钥协商公钥的一个参数，可以用户恢复出公钥对应的私钥。下面为 TCA 生成签名验证公钥 *Signature_Verification_TCertPub_key* 的算法。

（1） TCA 利用自己的密钥分发函数密钥 *TCA_KDF_Key* 及用户 Ecert 中的注册公钥 *EnrollPub_Key* 生成 *KeyDF_Key*:

$KeyDF\_Key= HMAC(TCA\_KDF\_Key, EnrollPub\_Key)$，

其中 HMAC(·)为 Hash 函数；

（2） TCA 利用 *KeyDF_Key* 和 *TcertIndex* 生成扩展值 *ExpansionValue*：

$ExpansionValue = HMAC(HMAC(KeyDF\_Key, “2”), TCertIndex)$，

其中 *TcertIndex* 为一个计数器的值，每生成一个证书，计数器的值加 2；

（3） TCA 利用扩展值 *ExpansionValue* 和注册公钥 *EnrollPub_Key* 生成签名验证公钥：

$Signature\_Verification\_TCertPub\_key= EnrollPub\_Key + ExpansionValue*G$，

其中 G 为椭圆曲线中的基点；

（4） TCA 生成 *TCertOwner_EncryptKey*：

$TCertOwner\_EncryptKey= [HMAC(KeyDF\_Key, “1”)]_{256\text{ - bit truncation}}$；

（5） TCA 生成 *TcertIndex* 密文：

$AES\_Encrypt_{TCertOwner\_EncryptKey}(TCertIndex\ ||常数)$。

TCA 产生密钥协商公钥的方法与产生签名验证公钥的方法类似，差别仅在于 *Key_Agreement_TCertPub_key* 中使用的 *TcertIndex'*为 *TcertIndex+1*。TCA 将批量产生的交易证书 Tcerts 及 *KeyDF_Key* 通过 TLS 安全通道发送给用户，用于用户恢复签名私钥及发起交易。

用户收到 TCA 发送的 Tcerts 及 *KeyDF_Key* 后，计算出 *TCertOwner_EncryptKey*，然后解密出 *TCertIndex*。*Signature_Verification_TCertPub_key* 对应的签名私钥为：

$TCertPriv\_Key = (EnrollPriv\_Key + ExpansionValue) modulo\ n$，

其中 $ExpansionValue = HMAC(HMAC(KeyDF\_Key, “2”), TCertIndex)$。密钥协商私钥的计算方法类似。

Fabric 系统中用户的交易可以分为两种类型：一种是需要知道交易者身份的，一种是不需要知道交易者身份的。当交易需要知道用户身份时，用户使用 Ecert 对应的私钥对交易进行签名，则交易可以直接追溯到用户身份。当交易不需要知道用户身份时，用户使用 Tcert 对应的私钥对交易进行签名。Fabric 上的一般节点是不能确定交易者身份的，而对于具有监管权限的节点可以追踪出用户的身份。此外，如果对于不同的交易使用不同 Ecert 对应的私钥签名，可以实现交易的不可链接性。对于交易的接收者而言，可以使用注册证书公钥或者交易证书公钥作为接收地址，实现交易接收的非匿名或匿名性。

在可审计方面，Fabric 上的任何获得 *TCA_KDF_Key* 的节点可以确认交易的发起者。审计节点利用 Fabric 系统上公布的用户注册证书中的 *EnrollPub_Key* 及 *TCA_KDF_Key* 产生 *KeyDF_Key*，进一步生成 *TCertOwner_EncryptKey*。对于 Fabric 系统中公布的交易信息及伴随的交易证书 Tcert，审计节点用 *TCertOwner_EncryptKey* 解密证书中的信息 $\text{AES\_Encrypt}_{TCertOwner\_EncryptKey}(TCertIndex \,||\, 常数)$，若常数正确则解密成功。审计节点获得正确的 *TCertIndex* 后，可以计算出 *ExpansionValue*，从而计算出用户注册证书公钥：

*EnrollPub_Key =Signature_Verification_TCertPub_key - ExpansionValue*G*,

因此可以确定交易者身份。

4 IDMixer 匿名隐私保护技术

IDMixer 是 IBM 提出的颁发匿名证书的方案[19],可用于交易的匿名认证及隐私保护等方面。Identity Mixer 的工作方式与传统公钥基础结构（PKI）中的客户端证书类似，但具有两个重要区别：

（1）灵活的公共密钥：用户可以拥有多个独立的公钥（假名）用于同一个密钥，而不是每个验证者绑定到固定单个公钥，从而为每个验证者甚至每个会话使用不同的假名。

（2）灵活的证书：证明用户属性的证书可以转换为用户任何假名的有效标记，这些标识只包含原始凭证中的一部分属性。转换后的标记在发行人的公开验证密钥下仍可验证。

IDMixer 的开源代码可以从 GitHub 获得。Github 上提供了关于如何构建开发环境的说明，以及如何使用帮助类将现有应用程序中的封装实体集成在一起的操作文档。其核心密码程序分别在专有许可下发布，允许商业和非商业用途。

在 IDMixer 方案中，用户可以自主选择想要出示的属性，且只需出示具体属性的相关假名（*Nym*）而不是实际值（比如证明自己大于 18 而不是具体岁数），其中假名（*Nym*）是一系列数值的承诺，类似于证书。假名之间是无关联的，只有用户可证明其中的秘密是相同的。同时 ID Mixer 能够防止无关方对不同认证凭证之间进行关联分析，并支持 CA 撤销功能。

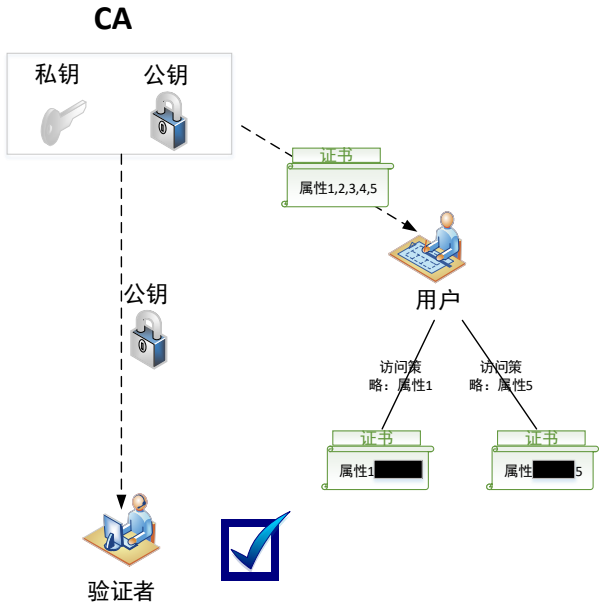

Fig.5 IDMixer scheme

图 5 IDMixer 方案

IDMixer 主要包含四个环节（如图 5 所示），

(1) 初始化（*Set up*）：证书权威（CA，Certificate Authority）运行初始化算法 *Setup*，计算

(Setcret key, public key)=Setup(secure param)，

产生签名密钥 *secret key* 并公布公钥 *public key*。

(2) 注册签发流程(*Enrollment/Issuance*)：用户 *User* 节点或客户端产生密钥 *sk* 并创建注册证书请求 *CertQst*，证书请求是对私钥 sk 的零知识证明，即 *CertQst=ZK{sk}*。然后用户利用证书请求 *CertQst* 在 CA 处注册自己的属性 *attr*，经过验证后 CA 对用户属性进行签名，

Cert=Sign(attr,secret key)，

产生一个电子证书 *Cert*，即用户属性以电子证书的形式签发（也称凭证或注册证书 *Cert*），该凭证存储在用户终端应用中。

(3) 签发展示环节(*Signing/Presentation*)：用户在访问不同应用时，访问控制策略指定了用户应在展示令牌中包含哪种类型的凭据和相应的属性。如果用户同意揭露访问所要求的政策信息，就从最初凭证（*Cert*）计算得到符合要求的展示令牌（*Sig*），

Sig=Sign(Cert,sk,m,Disclosure{attr})，

展示令牌实际是一个新的签名，包括：签署交易内容（*m*）；证明 CA 签发的注册证书（*Cert*）的有效性和所有权；揭露交易的访问控制策略所要求的属性（*Disclosure{attr}*）。最后用户将令牌（*Sig*）发给验证方（*Verify*）。

(4) 验证环节（*Verification*）：验证方 *Verifier* 使用 CA 的公钥 *public key* 验证令牌 *Sig* 是否符合访问控制策略。

5 Zcash 匿名隐私保护技术

Zcash 是最早以匿名隐私保护为核心的数字货币，在为用户提供交易地址的匿名的同时，可保证用户交易内容安全，其前身 ZeroCoin 是以比特币为交易货币的上层交易网络，实现比特币的交易匿名，但这种松耦合方

式一定程度上降低了系统的匿名性安全，因此 Zcash 团队于 2016 年 10 月 28 日发布了数字货币 Zcash[20][21]。

Zcash 的核心技术是 zk-SNARK（非交互式零知识证明，零知识证明技术的一种类型）技术，同时通过对已有的 zk-SNARK 的技术方案进行优化，使得 zk-SNARK 的验证过程和证据尺寸大大缩减，从而提升了实用性。其核心思想是对于 NP 困难问题 L，C 为规模 n 下的一个非确定性判断电路，一个 zk-SNARK 方式便是证明和验证 L 中的关系。详细来讲，给定规模 n，以电路 C 作为输入，一个可信方执行一次性初始化协议产生一个证明公钥 pk 和一个验证公钥 vk，证明公钥 pk 可以由任意方使用生成一个关于 $x \in L$ 的常量尺寸的证据 π，x 为其选择的一个规模 n 下的随机常量，任意方可使用验证公钥 vk 在线性时间内验证 π 的正确性。

Zcash 中的 zk-SNARK 算法核心包括三个部分：密钥生成、证据生成、证据验证，简要流程如下所示：

定义：C 为有限元 F 内的运算电路，$R_C=\{(x,a) \in F_n \times F_h : C(x,a)=0^l\}$，其中运算 C 的数学逻辑定义为 $L_C=(x \in Fn : \exists a \in F_h$,满足 $C(x,a)=0^l)$。

KeyGen$(1^\lambda, C) \rightarrow (pk, vk)$：设定安全参数 λ 和一个 F 有限域内的电路 C，概率性的抽样产生一个证据生成公钥 pk 和验证公钥 vk，这两个密钥作为公开参数用于产生/验证 L_C 中的关系。

Prove$(pk,x,a) \rightarrow \pi$:对于证据生成公钥 pk 和任意$(x,a) \in R_C$，证明者可针对于 $x \in L_C$ 产生一个非交互式证据 π。

Verify$(vk,x,\pi) \rightarrow b$:验证公钥 vk、输入 x 和证据 π，如果确定 $x \in L_C$ 则输出 b=1。

Zcash 通过使用 zk-SNARK 算法，将数字货币的交易过程分为：初始化(Setup)、产生地址(CreateAddress)、铸币(Mint)、打包交易(Pour)、验证交易(VerifyTransaction)、接收货币(Receive)六个流程，核心思路如下图所示：

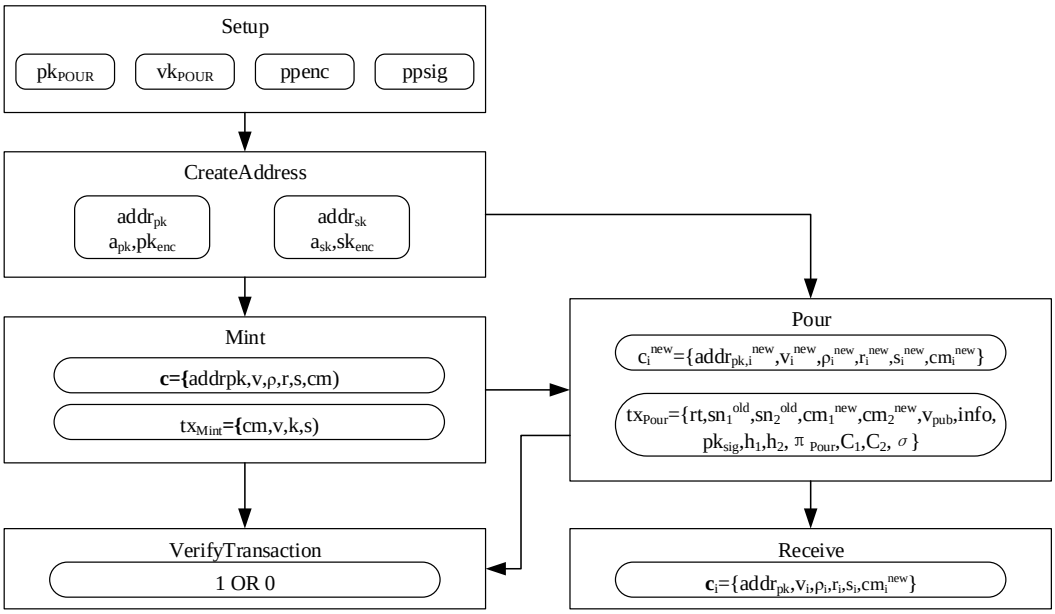

Fig.6 ZCash scheme

图 6 ZCash 方案

Setup 主要是产生系统的证据生成公钥和验证公钥，以及加密和签名算法的参数；CreateAdress 主要是产生交易地址，包括公钥地址和对应的私钥地址；Mint 主要是产生货币，**c** 为货币，tx_{Mint} 是需要向账本公开的铸币交易；Pour 是产生交易，c_i^{new} 为交易后的货币，tx_{pour} 是需要向账本公开的交易；VerifyTransaction 主要是对交易进行验证；Receive 是从 tx_{pour} 中获取货币。

6 典型隐私保护方案的比较

（1） 安全性比较分析

上述的五种方案，都可以实现交易的隐私性与可审计性，两者的区别如下：

a) 从交易发起者而言：CryptoNote 中的交易发起者利用自己的两对公私钥(a',A')和(b',B')派生出交易签名私钥 x'及签名验证公钥（P',R'），本地只需要存储两对公私钥，存储空间需求低，发起者利用交易接收者的公钥（A,B）派生出交易目标地址（P,R）；Monero 通过使用多个其他用户的交易公钥对用户交易公钥进行隐藏，确保了交易的匿名性；Fabric 中的交易发起者使用不同的 Tcert 对应的私钥对交易进行签名，利用交易接收者的交易证书中的公钥作为交易目标地址。为了实现交易的隐私性，对于不同的交易需使用不同的交易证书，证书量大，本地存储空间高。此外还需要 TCA 作为基础架构支持，TCA 负载重；ID Mixer 的交易发起者可根据交易接收者的具体要求出示带有不同属性（Disclosure{attr}）的证书 Sig，因此在交易中不会过多暴露用户的属性信息，可以有效保护交易发起者的身份隐私；Zcash 在发起交易的过程中使用 sn_1^{old} 和 sn_2^{old} 表示待交易货币，sn 作为货币序列号，只会在账本中出现过一次，表示该货币是未被交易过的，因此 Zcash 较好的保护了发起者的匿名性。

b) 从交易接收者而言：CryptoNote 中的交易接收者需要利用自己的私钥(a,b)遍历网络中广播的所有交易，确认自己是否为接收者，计算量大，效率低；Monero 需要根据自己存储的 I 值确认交易是否数据自己，计算量相对较大；Fabric 中的交易接收者很容易就能确认属于自己的交易，计算量低；ID Mixer 方案中交易接收方可以通过交易中公开的交易地址确认属于自己的交易，这种方式计算量低，但缺乏对交易接收方的匿名隐私保护；Zcash 中 tx_{pour} 中的 C_i 中携带有被交易者的公钥地址，但 C_i 是经过被交易者公钥加密密钥加密的，验证者无法获得 $addr_{sk,i}^{new}$，较好了保护了被交易者的匿名性。

c) 从监管审计而言：CryptoNote 中的审计节点需要执行和交易接收者相同的遍历计算才能确认交易的接收者与发送者，计算量大；Monero 不支持交易的监管审计；Fabric 中的审计节点需要利用不同的 TCertOwner_EncryptKey 解密证书中的信息 AES_EncryptTCertOwner_EncryptKey（TCertIndex || 常数），只有当解密后的常数正确时，才能确认交易的接收者，计算量也大。因此，CryptoNote 和 Fabric 方案中的审计功能代价都比较高；ID Mixer 和 Zcash 也不支持监管审计 。

d) 从系统安全性而言：CryptoNote 方案中的用户可以使用非监管的密钥进行交易，逃脱审计；Monero 不支持监管审计；Fabric 中的每笔交易都需要使用 TCA 颁发的证书，不能逃脱审计；ID Mixer 和 Zcash 均不支持监管，因此也不存在逃脱监管审计之说，但 Zcash 在初始化时会掌握整个系统的最高权限，如果攻击者一旦掌握了 Zcash 初始化参数，则可以在系统中制造任意数量的货币和交易，存在极大的安全风险。

综上所述，CryptoNote 和 Fabric 方案均是使用传统的密码算法通过对密钥的派生算法进行优化实现可监管的匿名隐私保护，CryptoNote 简单但无法强制监管，Fabric 可强制监管但每次交易均需要 TCA 颁发交易证书，Monero 通过环签名实现交易的匿名化处理。ID Mixer 和 Zcash 均是利用零知识证明技术实现交易的高强度匿名隐私保护，但 ID Mixer 只针对属性信息进行隐藏，使用范围有限，Zcash 可实现交易身份、交易金额的匿名隐藏，但系统初始化时权限过高，存在较大安全威胁。

（2） 效率比较分析

首先从运算效率方面对方案进行比较分析。

在本次效率比较分析中，选择椭圆曲线上的群，阶数为 170bit 的大质数，安全性与 1024bit 的 RSA 算法等价。定义 E 为指数运算，H 为哈希运算，P 为双线性对运算，k 为全部属性个数，l 为隐藏属性个数。忽略乘法、加法运算。

CryptoNote 方案中，生成用户密钥需要 2 次指数运算，不存在证书请求、证书请求验证，生成交易密钥算法中需要 3 次指数运算和 1 次哈希运算，签名阶段使用 SM2 算法进行签名，共包括 1 次指数运算和 1 次哈希运算，验证阶段使用 SM2 算法进行验证，共包括 3 次指数运算和 1 次哈希运算，监管阶段需要对系统中存在的公钥进行遍历，每次遍历需要使用 2 次指数运算和 1 次哈希运算，假设遍历 m 次得到结果。

Monero 方案，生成用户密钥需要 2 次指数运算和 1 次哈希运算，不存在证书请求、证书请求验证，生成交易算法中只要将其他用户的交易公钥用来打包成交易集合即可，签名阶段需要使用 8n 次指数运算和 n+1 次哈希运算，其中 n 为交易集合中公钥的个数，验证阶段需要 5n 次指数运算和 n+1 次哈希运算，不存在监管。

Fabric 匿名隐私保护方案中，生成用户密钥需要 1 次指数运算，生成交易密钥需要 2 次指数运算和 3 次哈希运算，签名和验证使用 SM2 算法，同 CryptoNote，监管使用查询数据库方式进行，但需要进行 SM4 算法解密。

在 idmixer 算法中，生成密钥算法需要(k+11)次指数运算和 2 次哈希运算，生成证书请求需要进行 4 次指数运算和 1 次哈希运算，证书请求运算验证需要 3 次指数运算和 1 次哈希运算，生成交易公钥需要进行(k+2)次指数运算，签名需要进行(k+10)次指数运算，2 次哈希运算和 2 次双线性对运算，方案计算开销对比结果如表 1 所示。

Table 1 Computation costs of Identity Mixer scheme

表 1 Identity Mixer 方案的计算开销

	CryptoNote	Monero	Fabric	Identity Mixer 方案
生成密钥	2E	2E+H	E	(k+11)E+2H
证书请求	-	-	-	4E+H
证书请求验证	-	-	-	3E+H
生成交易公钥	3E+H	-	2E+3H	(k+2)E
签名	E+H	8nE+(n+1)H	E+H	(l+14)E+2H
验证	3E+H	5nE+(n+1)H	3E+H	(k+10)E+2H+2P
监管	m(2E+H)	-	-	-

其次从空间效率方面对方案进行分析。

CryptoNote 方案中，不存在 CA，用户私钥长度为 2 个大整数，即 2*170bit，公钥长度为 2 个椭圆曲线上的点，即 2*2*170bit，交易公钥为 2 个椭圆曲线上的点，即 2*2*170bit，签名使用 SM2 算法进行签名，则签名长度为 2 个大整数，即 2*170bit，监管密钥为 1 个大整数和 1 个椭圆曲线上的点，即 170+2*170bit。

Monero 方案中，不存在 CA，用户私钥长度为 1 个大整数，即 170bit，公钥长度为 2 个椭圆曲线上的点，即 2*2*170bit，交易公钥为 n 个椭圆曲线上的点，即 n*2*170bit，签名结果为(2*n+2)*170bit。

Fabric 匿名隐私保护方案中，CA 私钥长度为 170bit，用户私钥长度为 170bit，公钥长度为 2*170bit，交易公钥长度为 2*170bit，签名同 CryptoNote，监管密钥长度为对称私钥，长度为 128bit。

Idmixer 方案中，CA 私钥和用户私钥长度均为 170bit，公钥证书长度为(k+5)170+340bit，交易公钥长度（证书长度）为 682bit，签名长度（出示令牌长度）为 170l+2044bit。

Table 2 Storage Cost of Identity Mixer scheme costs

表 2 Identity Mixer 方案的存储开销

	CryptoNote	Monero	Fabric	Identity Mixer 方案
CA 私钥长度(bit)	-	-	170	170
用户私钥长度(bit)	2*170	170	170	170
公钥长度(bit)	2*2*170	2*2*170bit	2*170	(k+5)170+340
交易公钥长度(bit)	2*2*170	n*2*170bit	2*170	682
出示令牌长度(bit)	2*170	(2*n+2)*170bit	2*170	170·l+2044
监管密钥长度(bit)	170+2*170	-	128	-

由于 Zcash 中涉及到的密码运算较为复杂，此处对于 Zcash 的效率分析不使用基于密码学原语的效率分析，而是通过计算机进行模拟，给出模拟数据。Zcash 方案选择 Alt_BN128 曲线，循环群阶数为 256bit 的大

素数，在 Intel Core i7-4770 @3.4GHz，16GB of RAM PC 环境下运行效率和占用存储空间如表 3 和表 4 所示所示。

Table 3 Running efficiency of Zcash scheme

表 3 Zcash 方案的运行效率

	时间
Setup	5min 17s
CreateAddress	326ms
Mint	23us
Pour	2min 2s
Verify Transaction	8.3us(mint) 5.7ms(pour)
Receive	1.6ms

如表三所示，Zcash 在初始化（Setup），创建账户（CreateAddress）和花币（Pour）过程中耗时较多，Pour 交易验证及接收代币（Receive）阶段耗时很少，发币（Mint）和发币验证阶段耗时几乎可以忽略。

Table 4 storage costs of Zach Scheme

表 4 Zcash 方案的存储开销

	存储量
公开参数 pp	896MB
公钥地址 $addr_{pk}$	343B
私钥地址 $addr_{sk}$	319B
Zcash 货币 c	463B
Mint 交易 tx_{Mint}	72B
Pour 交易 tx_{Pour}	996B

如表 4 所示，Zcash 初始化产生的公共参数需要占用较多的存储空间，用户交易地址和交易占用存储量较小。综上可知，Zcash 初始化效率较低，存储量较大，对中心服务器性能要求较高，在运行效率和存储开销上仍存在改进的空间。

7 总结

本文对典型的区块链匿名隐私保护技术进行了比较分析，主要包括 CryptoNote 匿名隐私保护技术、Monero 匿名隐私保护技术、Fabric 匿名隐私保护技术、IDMixer 匿名隐私保护技术、Zcash 匿名隐私保护技术。其中 CryptoNote 简单易用，为用户提供匿名保护，但不能实现交易数据保护；Monero 简单易用，但尺寸相对较大，不能有效监管；Fabric 通过引入 TCA 实现交易证书的颁发，既实现了匿名又可进行数据隐私保护，但每次交易证书均需 TCA 重新颁发；IDMixer 将身份属性信息认证与零知识证明结合，解决了 CA 管理中的匿名隐私保护，但局限性较大；Zcash 不仅实现了交易匿名处理，同时实现了交易金额的密文比较和验证，但执行效率较低和尺寸较大。目前学术界针对于 Zcash 的优化以及其基础核心零知识证明技术的研究正在如火如荼的开展，不断有较新的研究成果提出，也为区块链匿名隐私保护技术的发展和落地提供了有利的支撑。

References:

[1] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. 2008.

[2] 赵刚.《区块链价值互联网的基石》,电子工业出版社, 2016.

[3] 新华网，2018.5.28. http://www.xinhuanet.com/2018-05/28/c_1122901308.htm.

[4] 邹均, 张海宁, 唐屹, 李磊等. 《区块链技术指南》, 机械工业出版社, 2017.

[5] 中国互联网金融协会成立区块链研究工作组. http://finance.sina.com.cn/chanjing/cyxw/2016-06-16/doc-ifxtfrrf0468084.shtml.

[6] Factom: A Scalable Data Layer for the Blockchain. http://factom.org/.

[7] 区块链在互联网保险中应用. http://www.midai.cn/sound/50224.

[8] 巨头英特尔宣布在游戏中尝试全新研发的区块链技术. http://mt.sohu.com/20160227/n438725218.shtml.

[9] Tatsuaki Okamoto and Kazuo Ohta. Universal electronic cash. In CRYPTO, pages 324–337, 1991.

[10] Fergal Reid and Martin Harrigan. An analysis of anonymity in the bitcoin system. CoRR, abs/1107.4524, 2011.

[11] Dorit Ron and Adi Shamir. Quantitative analysis of the full bitcoin transaction graph. IACR Cryptology ePrint Archive, 2012:584, 2012.

[12] Micha Ober, Stefan Katzenbeisser, and Kay Hamacher. Structure and anonymity of the bitcoin transaction graph. Future internet, 5(2):237–250, 2013.

[13] Marc Santamaria Ortega. The bitcoin transaction graph — anonymity. Master's thesis, Universitat Oberta de Catalunya, June 2013.

[14] Nicolas van Saberhagen. CryptoNote v 2.0, https://cryptonote.org/whitepaper.pdf, 2013-10-17.

[15] Monero, https://getmonero.org/.

[16] Eiichiro Fujisaki. Sub-linear size traceable ring signatures without random oracles. In CT-RSA, pages 393-415, 2011.

[17] Ronald Cramer, Ivan Damgard, and Berry Scheoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In CRYPTO, pages 174-187, 1994.

[18] https://github.com/hyperledger-archives/fabric/blob/master/docs/protocol-spec.md#4.-Security, 2016.

[19] Identity Mixer, http://idemix.wordpress.com/.

[20] Zcash. https://z.cash/.

[21] Sasson E B, Chiesa A, Garman C, et al. Zerocash: Decentralized Anonymous Payments from Bitcoin[C]// Security and Privacy. IEEE, 2014:459-474.