基于门限环签名的可删除区块链

任艳丽[1]，徐丹婷[1]，张新鹏[1]，谷大武[2]
（1. 上海大学通信与信息工程学院，上海 200444；
2. 上海交通大学电子信息与电气工程学院，上海 200240）

摘 要：随着区块链的发展，存储所有区块数据需要巨大的存储空间，而数据一旦写入链中就不能更改，可能会造成过期数据占用大量存储空间的问题。首先对门限环签名方案进行改进，然后基于空间证明的共识机制提出了可删除的区块链。当某个区块数据过期或失效时，经大多数节点同意并签名后，可对该区块进行有效删除，并保持区块链的总体结构不变。在模拟环境中进行了仿真实验，结果表明，所提区块链方案在生成和删除区块时效率都很高，且不影响其他区块的存储和使用。

关键词：区块链；数据可删除；门限环签名；空间证明
中图分类号：TP309
文献标识码：A
doi: 10.11959/j.issn.1000−436x.2019084

Deletable blockchain based on threshold ring signature

REN Yanli[1], XU Danting[1], ZHANG Xinpeng[1], GU Dawu[2]
1. School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China
2. School of Electronic Information and Electrical Engineering, Shanghai Jiaotong University, Shanghai 200240, China

Abstract: With the development of blockchain, huge storage space is needed to store all of blockchain data. In addition, data can't be changed once it is packaged into the chain, and it may be possible for overdue data to take up large storage space. Firstly, a threshold ring signature scheme was improved, and then a deletable blockchain scheme based on the mechanism of proof of space (PoSpace) consensus was constructed. Most of nodes could sign and delete a block together when block data was overdue, and the overall structure of the blockchain was unchanged. Several experiments in a simulated environment were executed, and the results show that the proposed blockchain scheme has high efficiency in generating and delegating a block. Meanwhile, the deletion of a block doesn't influence the storage and use of other blocks.

Key words: blockchain, data deletable, threshold ring signature, proof of space

1 引言

区块链技术[1]运用数据加密、时间戳、分布式共识和经济激励等手段，在节点不需要互相信任的分布式系统中实现去中心化的点对点交易，同时能按照时间顺序将交易相关数据以数据区块的形式存储，其中，数据区块以链条的方式组合成特定数据结构，并以密码学原理保证数据不可篡改和不可伪造，形成去中心化的共享总账。区块链技术以其区别于中心化系统的核心优势，引起了人们的广泛关注，拥有广泛的应用前景[2]。

共识机制是区块链实现去中心化的核心内容，其中基于工作量的证明（PoW, proof of work）[1,3]和基于权益的证明（PoS, proof of stake）[4-5]是最为普遍的区块链共识机制。但是，PoW 会造成大量的电力能源损耗，且挖矿过程中产生的算力无法应用于别处，是一种纯粹的算力损耗；而 PoS 挖矿成本低，容易受到攻击，且挖矿收益与权益的相互促进，

收稿日期：2018–07–30；**修回日期**：2019–02–26
基金项目：国家自然科学基金资助项目（No.U1736120, No.61572309, No.61525203, No.U1636206）
Foundation Item: The National Natural Science Foundation of China (No.U1736120, No.61572309, No.61525203, No.U1636206)

使系统日益趋于中心化。2017 年，Park 等[6]提出了基于空间证明的共识机制（PoSpace, proof of space），通过存储空间的竞争来获得记账权，即竞争节点分别给出自身存储空间大小的证明，验证通过后空间大的节点生成新的区块。同 PoW 和 PoS 体制相比，PoSpace 基于空间证明，不需要复杂的计算，大大降低了能源消耗，提高了共识效率，且节点的磁盘空间在证明结束后可得到释放，用于下一次竞争，实现了资源的循环使用。

匿名性是区块链交易中实现用户隐私保护的基本要求，而环签名对数据进行认证的同时可确保签名者的匿名性，是区块链研究中的重要工具。环签名是 Rivest 等[7]在 2001 年提出的，允许某个成员代表一组用户进行签名，而不泄露签名用户的身份。随后，Bresson 等[8]提出了门限环签名，即环中的部分成员只要达到规定的门限值，就可以代表环中所有用户进行签名，该方案提出了公平拆分的思想，并在随机预言模型中可证安全。当签名人数较少时，方案是高效的，但是当签名人数较多时，方案效率较低。Toshiyuki 等[9]针对签名人数较多的情况，提出了高效的门限环签名方案，但该方案存在安全问题，详细分析请见本文第 3 节。Chung 等[10]使用双线性对映射，提出了基于身份的门限环签名方案，但效率不高。以上方案都是基于传统的密码学困难问题，如陷门单向函数、离散对数问题等。2011 年，Melchor 等[11]基于编码理论，提出了高效的门限环签名方案。Zhang 等[12]基于 MQ（multivariate quadratic）问题，提出了抗量子攻击的门限环签名方案。

随着区块链的发展，存储所有区块数据需要巨大的存储空间，对于普通节点来说存储代价很大。而在实际应用中，有的区块数据已经过期或失效，比如已经宣布破产的银行交易信息、已经废除的法律文件等。这些数据永久存储在区块链中已经失去了价值，且占用大量空间，造成资源浪费。如果经大多数用户同意后，可删除这些失效的数据，而又不影响区块链中其他数据的存储和使用，有助于释放节点的存储空间，降低存储代价，具有重要的应用价值。目前绝大多数区块链结构是不允许删除数据的，数据一旦写入链中就不能更改，可能会造成过期数据占用大量存储空间的问题。

在区块链研究中，爱哲森公司基于变色龙散列，提出了可编辑区块链技术[13]。只要知道变色龙散列的陷门，就可以找到已有数据的一个碰撞，从而实现对区块链数据的编辑。但在该方法中，散列函数的陷门存储在一个可信中心，因此对数据的编辑权过于中心化。Li 等[14]使用秘密共享技术，将变色龙散列的陷门分成多份，并分配给各个网络节点。当一半以上的网络节点同意时，可恢复散列函数的陷门，找到已有数据的碰撞，对区块链数据进行修改。以上 2 种方法均使用变色龙散列，而目前的区块链结构大多使用抗碰撞的散列函数，因此上述方法的适用范围非常有限。基于抗碰撞的散列函数，本文提出了可删除的区块链方案，适用于大多数区块链结构，具有广阔的应用前景。

本文首先分析了文献[9]中的门限环签名方案，指出了该方案的安全问题，提出了改进方案，并在随机预言模型中可证安全。随后，使用改进的门限环签名方案，基于空间证明的共识机制提出了可删除的区块链。当数据过期或失效时，经大多数用户同意并签名，对过期数据进行删除，并保持区块链的结构不变，不影响其他区块的存储和使用。所有人可对数据进行验证，包括被删除数据的相关信息。

2 基础知识

2.1 门限环签名

环签名[7]主要包括以下算法。

setup 算法：输入安全参数 λ，输出 n 个环成员的公钥 $\mathrm{PK}_1,\cdots,\mathrm{PK}_n$ 和私钥 $\mathrm{SK}_1,\cdots,\mathrm{SK}_n$。

ring-sign 算法：输入消息 m、n 个环成员公钥 $\mathrm{PK}_1,\cdots,\mathrm{PK}_n$ 和签名用户私钥 SK_S，输出环签名 σ。

ring-verify 算法：输入环成员公钥 $\mathrm{PK}_1,\cdots,\mathrm{PK}_n$ 和签名 (m,σ)，输出“接收”或者“拒绝”。

安全的环签名需要满足正确性、不可伪造性以及匿名性，即合法环签名可以验证通过，非环成员不能伪造合法的环签名，且任何人不能获得环签名用户的真实身份。

在此基础上，门限环签名[8]包括以下算法。

setup 算法：输入安全参数 λ，输出 n 个环成员的公钥 $\mathrm{PK}_1,\cdots,\mathrm{PK}_n$ 和私钥 $\mathrm{SK}_1,\cdots,\mathrm{SK}_n$。

T-ring-sign 算法：输入消息 m、n 个环成员公钥 $\mathrm{PK}_1,\cdots,\mathrm{PK}_n$ 和 $n-t$ 个签名用户私钥 $\mathrm{SK}_{i,1},\cdots,\mathrm{SK}_{i,n-t}$，输出环签名 σ。

T-ring-verify 算法：输入环成员公钥 $\mathrm{PK}_1,\cdots,\mathrm{PK}_n$ 和签名 (m,σ)，输出“接收”或者“拒绝”。

本文所提环签名方案中用到了文献[7]中提出的用户拆分方法，下面做简要介绍。

定义 1　公平拆分[7]。假设系统中 n 个用户被拆分为 t 个子集 $\pi=(\pi^1,\cdots,\pi^t)$，$I=\{i_1,\cdots,i_t\}$ 表示其中 t 个用户编号的集合。对于集合 I，如果对于所有的 $j\in[1,t]$，均有 $\#(I\bigcap\pi^j)=1$，就称 π 是一个公平拆分，其中#(X)表示集合 X 中元素的个数。

如文献[7]所述，为了确保环签名中用户的匿名性，对于任意包含 t 个用户的集合，都要存在一个公平拆分，称为完备拆分系统。具体定义如下。

定义 2　完备拆分系统[7]。令 $t<n-t<n$，对于任意包含 t 个用户编号的集合 I，都存在一个公平拆分，这样的系统称为(n, t)–完备拆分系统。

2.2　环签名安全模型

本节介绍门限环签名方案的安全模型，即选择消息攻击下的强不可伪造性（SU-TRS-CMA, strong unforgeability against chosen message attack in a threshold ring signature scheme）。

定义 3　SU-TRS-CMA。假设环中共有 n 个成员，其中 $n-t$ 个成员可代表所有用户签名。通过以下游戏，本文定义 TRS（threshold ring signature）方案在选择消息攻击下的强不可伪造性[8,15]。游戏包含 2 个参与者：挑战者和敌手。具体包括以下步骤。

setup　挑战者执行 setup 算法获得所有环成员公钥 $\mathrm{PK}_1,\cdots,\mathrm{PK}_n$ 和私钥 $\mathrm{SK}_1,\cdots,\mathrm{SK}_n$，并发送公钥给敌手。

阶段 1　敌手适应性地进行下列询问。

散列询问：敌手发送字符串给挑战者，挑战者选择随机数作为对应的散列值，并返还给敌手。

私钥询问：敌手发送用户公钥给挑战者，挑战者返回对应私钥给敌手，敌手至多可询问 $n-t-1$ 个用户的私钥。

签名询问：敌手发送消息 m 给挑战者。挑战者随机选择 $n-t$ 个环成员私钥 $\mathrm{SK}_{i,1},\cdots,\mathrm{SK}_{i,n-t}$，根据 T-ring-sign 算法生成签名 σ，并返回给敌手。

挑战阶段　假设敌手能以不可忽略的概率 ε 伪造有效的门限环签名，并将消息 m^* 的环签名 σ^* 发送给挑战者，且 (m^*,σ^*) 没有出现在签名询问阶段。挑战者可根据 (m^*,σ^*)，以不可忽略的概率 ε' 解决单向函数的求逆问题。

在上述游戏中，敌手的优势定义为 ε，即成功伪造门限环签名的概率。如果对于任意 t 多项式时间的敌手，在经过 q 次询问后，至多能以概率 ε 伪造有效的环签名，则称门限环签名方案在选择消息攻击下是(t,q,ε)–强不可伪造的。

由以上安全模型，如果门限环签名方案满足强不可伪造性，则攻击者即使得到消息 m 的签名 σ，也不能伪造 m 的其他有效签名 $\bar{\sigma}$，且 $\bar{\sigma}\neq\sigma$。

2.3　基于空间证明的区块链

PoSpace 共识机制[6]基于“一定时间内需要一定空间构造一个结构图”这一事实，展开空间竞争。首先，介绍空间证明过程。如图 1 所示，设定有向无环结构图 $G=(V,E)$，其中 $V=\{v_1,v_2,\cdots,v_N\}$ 为顶点集合，N 为顶点个数，E 为有向边集合。图中各顶点因为有向边而存在一定的联系，为进一步表示顶点之间的联系，突出该图的结构，为每个顶点设置与之相关联的标签值，具体为

$$l_i=\mathrm{hash}(\mu,i,l_{p_1},l_{p_2},\cdots,l_{p_t}),i=1,2,\cdots,N$$

其中，i 为顶点序号，μ 为可设定的随机数，$l_{p_1},l_{p_2},\cdots,l_{p_t}$ 为链向当前顶点 i 的所有前向顶点，即其母顶点的标签值。这样，每个顶点与其母顶点便依靠标签值联系起来，形成了基于顶点标签值的有向图结构。

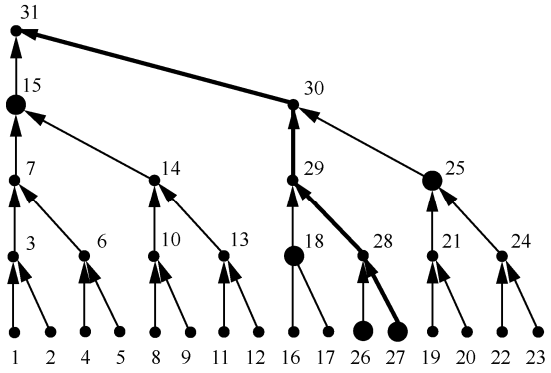

图 1　有向无环结构图

从图 1 可以看出，要实现对以上结构图的存储需要一定的空间，空间越大，越容易实现结构图的存储与恢复。而在空间不足时，只能多次利用仅有的空间存储相关数据，反复存储与删除，以时间换取空间。因此当时间一定时，空间大小不同的矿工对同一结构图的存储效率便高低不齐，产生基于空间的竞争。PoSpace 正是在这样的模型下建立的。

PoSpace 共识系统会生成如图 1 所示的结构图，

每个矿工都可以依据自己的空间大小，最大程度并最优化地存储该结构图。系统会选择存储空间最大的矿工作为最终的记账者。因此，基于空间证明的挖矿过程就是矿工证明自己拥有足够大的空间，以便高效存储结构图的过程。矿工拥有的空间越大，挖矿成功的概率也越大。挖矿的具体实现过程可参考文献[6]。

接下来，介绍基于空间证明的区块链结构，如图 2 所示。

由图 2 可知，每个区块 i 都分为 3 个子块：证明子块 φ_i（hash φ_i）、签名子块 σ_i（signature σ_i）和交易子块 τ_i（transaction τ_i）。其中，证明子块又称散列子块，是对相关内容做散列运算后的结果。

1) 散列子块 φ_i 包含：当前区块序号 i，记账者对前一区块的散列子块 φ_{i-1} 的签名 ζ_φ，记账者在竞争记账权时给出的承诺证明以及空间证明 $(p_{k_i},\gamma_i,c_i,a_i)$。即 $\varphi_i=\mathrm{hash}(i,\zeta_\varphi,(p_{k_i},\gamma_i,c_i,a_i))$。

2) 签名子块 σ_i 包含：当前区块序号 i，记账者对当前区块的交易子块 τ_i 的签名 ζ_τ，记账者对前一区块的签名子块 σ_{i-1} 的签名 ζ_σ。即 $\sigma_i=\{i,\zeta_\tau,\zeta_\sigma\}$。

3) 交易子块 τ_i 包含：当前区块序号 i，交易信息列表 ctx。即 $\tau_i=\{i,\mathrm{ctx}\}$。

3 改进的门限环签名方案

3.1 IT-TRS 方案及安全性分析

首先，回顾一下 IT-TRS（TOSHIYUKI I, KEISUKE T- threshold ring signature）方案[9]。

假设系统中有 n 个用户，其中 $n-t$ 个用户可代表所有用户生成环签名，其余 t 个用户不参与签名。令 $P_{i_1},\cdots,P_{i_t}$ 表示非签名用户，而 $I=\{i_1,\cdots,i_t\}$ 表示非签名用户编号的集合。假设通过拆分，所有用户被拆分为 t+1 个互不相交的子群，则至少有一个子群中的用户都参与了签名，称这个子群为合法子群。如文献[8]所述，为了确保环签名的不可伪造，需要采用$(n,t+1)$−完备拆分系统。

setup 算法：令 l 表示安全参数，$\prod_n^{t+1}=\{\pi_1,\cdots,\pi_p\}$ 是一个$(n,t+1)$−完备拆分系统，其中，$p=2^{t+1}\mathrm{lb}n$，π_i 表示一次公平拆分，记 $\pi_i=\{\pi_i^1,\cdots,\pi_i^{t+1}\}$，$\pi_i^j$ 表示一组用户编号的集合。令 $P_1,\cdots,P_n$ 表示所有 n 个用户。对于每一个 $i=1,2,\cdots,n$，g_i 表示匿名的单向陷门映射。对于任意 i 和 j，q_i^j 表示 π_i^j 中元素的个数，Q 表示 q_i^j 的最大值。令 $\pi_i^j=\{p_i^{j,1},\cdots,p_i^{j,q_i^j}\}$。如果对于所有的 $k\in\pi_i^j$，P_k 都是签名者，称 π_i^j 是合法的。如文献[8]所述，对于所有的整数 n 和 t，且 $t\leqslant n$，$(n,t+1)$−完备拆分系统一定是存在的，每个用户 P_i 基于单向陷门映射 g_i 进行签名。

对于每个子群 π_i^j，定义单向陷门映射为

$$G_i^j:G_i^j(x_1,\cdots,x_Q)=g_{p_i^{j,1}}(x_1)\|\cdots\|g_{p_i^{j,Q}}(x_Q)$$

如果 $q_i^j=Q$，令 $S_i^j=\pi_i^j$，否则

$$S_i^j=\{\pi_i^j\cup\{p_i^{j,q_i^j+1},p_i^{j,q_i^j+2},\cdots,p_i^{j,Q}\}\}$$

其中，$p_i^{j,q_i^j+1}=p_i^{j,q_i^j+2}=\cdots=p_i^{j,Q}=p_i^{j,q_i^j}$，$G_i^j$ 的陷门是所有 $p_i^{j,k}$ 陷门的集合。

T-ring-sign 算法：假设系统中至多有 t 个用户不参与签名。由于所有用户被分为 t+1 个互不相交的子群，则每次拆分都存在一个合法子群，记为 S_i^j。设签名消息为 m，H 为输出长度为 Ql bit 的 hash 函数。对每个 $\pi_i,i=1,2,\cdots,p$，签名者执行以下步骤。

步骤 1 随机选择 $s^1,\cdots,s^Q\in\{0,1\}^l$，计算 $v_{j+1}=H(m,s^1,\cdots,s^Q)$。

步骤 2 对于 $k=j+1,\cdots,t+1,1,2,\cdots,j-1$，随机选择 $x_k^1,\cdots,x_k^Q\in\{0,1\}^l$，计算 $v_{k+1}=H(m,v_k\oplus G_i^k(x_k^1,\cdots,x_k^Q))$，其中，$v_{t+2}=v_1$。

步骤 3 签名者使用 G_i^j 的陷门求出 $x_j^1,\cdots,x_j^Q$，使 $(s^1,\cdots,s^Q)=v_j\oplus G_i^j(x_j^1,\cdots,x_j^Q)$，则 $v_{j+1}=H(m,$

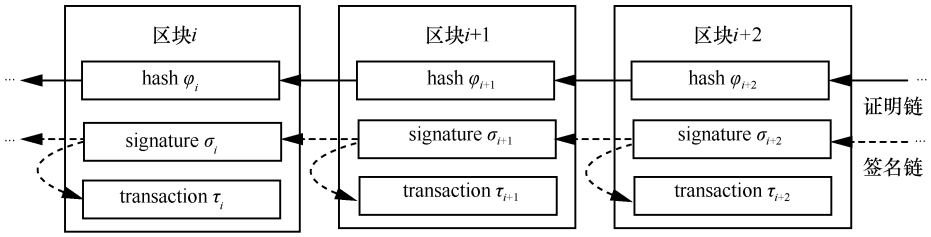

图 2 基于空间证明的区块链结构

$v_j \oplus G_i^j(x_j^1,\cdots,x_j^Q))$。

步骤 4　随机选择 $i_0 \in \{1,2,\cdots,t+1\}$，则消息针对集合 π_i 的签名为

$$\sigma_i = (PK_{p_i^{1,1}},\cdots,PK_{p_i^{1,Q}},PK_{p_i^{2,1}},\cdots,PK_{p_i^{t+1,Q}};$$
$$i_0,v_{i_0};x_1^1,\cdots,x_1^Q,x_2^1,\cdots,x_{t+1}^Q)$$

因此，消息 m 的最终签名 $\sigma=(\sigma_1,\cdots,\sigma_p)$。

T-ring-verify 算法：对于签名 σ_i，验证者计算

$$v_{k+1}=H(m,v_k \oplus G_i^k(x_k^1,\cdots,x_k^Q)),$$
$$k=i_0,\cdots,t+1,1,2,\cdots,i_0-2$$

其中，$v_{t+2}=v_1$，并验证 $v_{i_0}=H(m,\ v_{i_0-1}\oplus G_i^{i_0-1}(x_{i_0-1}^1,\cdots,x_{i_0-1}^Q))$ 是否成立。

若所有的 $\sigma_i(i=1,2,\cdots,p)$ 都能验证通过，则 σ 是消息 m 的有效签名。

通过分析，上述方案中的门限环签名不满足强不可伪造性。假设合法子群 π_i^j 按照上述算法生成消息 m 的签名 σ，则非合法子群 π_i^{j+1} 可与合法子群中的某个用户相勾结，生成消息 m 的另外一个合法签名 $\overline{\sigma}$。不妨设非合法子群 π_i^{j+1} 与合法子群 π_i^j 中的用户 $p_i^{j,1}$ 相勾结，攻击过程如下。

步骤 1　计算 $(s^1,\cdots,s^Q)=v_j\oplus G_i^j(x_j^1,\cdots,x_j^Q)$，并重新选择 $\overline{s^1}\in\{0,1\}^l$，计算 $\overline{v_{j+1}}=H(m,\overline{s^1},s^2,\cdots,s^Q)$。

步骤 2　非合法子群 π_i^{j+1} 使用 G_i^{j+1} 的陷门求出 $\overline{x_{j+1}^1},\cdots,\overline{x_{j+1}^Q}$，使 $\overline{v_{j+2}}=H(m,\overline{v_{j+1}}\oplus G_i^{j+1}(\overline{x_{j+1}^1},\cdots,\overline{x_{j+1}^Q}))=v_{j+2}$。

步骤 3　用户 $p_i^{j,1}$ 使用 $g_{p_i^{j,1}}(\cdot)$ 的陷门求出 $\overline{x_j^1}$，使 $(\overline{s^1},\cdots,s^Q)=v_j\oplus G_i^j(\overline{x_j^1},\cdots,x_j^Q)$。

步骤 4　随机选择 $\overline{i_0}\in\{1,2,\cdots,t+1\}$，则消息 m 针对集合 π_i 的签名为

$$\overline{\sigma_i}=(\mathrm{PK}_{p_i^{1,1}},\cdots,\mathrm{PK}_{p_i^{1,Q}},\mathrm{PK}_{p_i^{2,1}},\cdots,\mathrm{PK}_{p_i^{t+1,Q}};\overline{i_0},v_{i_0};$$
$$x_1^1,\cdots,x_1^Q,\cdots,\overline{x_j^1},x_j^2,\cdots,x_j^Q,\overline{x_{j+1}^1},\cdots,\overline{x_{j+1}^Q},\cdots,x_{t+1}^Q)$$

把以上过程重复 p 次，生成最终的签名 $\overline{\sigma}=(\overline{\sigma}_1,\cdots,\overline{\sigma}_p)$。

由验证算法可知，上述签名可以通过验证。攻击成功的原因在于随机数 s^k 的改变只影响 x_j^k 的值，而对 $x_j^1,\cdots,x_j^{k-1},x_j^{k+1},\cdots,x_j^Q$ 没有影响，因此攻击者只要知道 G_i^j 的某个陷门，就可以伪造签名通过验证。

3.2　改进的 TRS 方案

针对上述攻击方法，本文提出了改进的 TRS（I-TRS, improved threshold ring signature）方案，满足门限环签名的强不可伪造性及用户的匿名性。

假设系统中有 n 个用户，其中 n–t 个用户可代表所有用户生成环签名，其余 t 个用户不参与签名。改进方案中的数学符号与原方案相同。

setup 算法：同原方案。

T-ring-sign 算法：假设系统中至多有 t 个用户不参与签名。由于所有用户被分为 t+1 个互不相交的子群，则每次拆分都存在一个合法子群，记为 S_i^j。设签名消息为 m，H 为输出长度为 Ql bit 的 hash 函数。对每个 $\pi_i, i=1,2,\cdots,p$，签名者执行以下步骤。

步骤 1　随机选择 $s^1,\cdots,s^Q\in\{0,1\}^l$，计算 $v_{j+1}=H(m,H(s^1,\cdots,s^Q))$。

步骤 2　对于 k=j+1,$\cdots$,t+1,1,2,$\cdots$,j−1，随机选择 $x_k^1,\cdots,x_k^Q\in\{0,1\}^l$，计算 $v_{k+1}=H(m,v_k\oplus G_i^k(x_k^1,\cdots,x_k^Q))$，其中 $v_{t+2}=v_1$。

步骤 3　签名者使用 G_i^j 的陷门求出 $x_j^1,\cdots,x_j^Q$，使 $H(s^1,\cdots,s^Q)=v_j\oplus G_i^j(x_j^1,\cdots,x_j^Q)$，则 $v_{j+1}=H(m,v_j\oplus G_i^j(x_j^1,\cdots,x_j^Q))$。

步骤 4　随机选择 $i_0\in\{1,2,\cdots,t+1\}$，则消息 m 针对集合 π_i 的签名为 $\sigma_i=(\mathrm{PK}_{p_i^{1,1}},\cdots,\mathrm{PK}_{p_i^{1,Q}},\mathrm{PK}_{p_i^{2,1}},\cdots,\mathrm{PK}_{p_i^{t+1,Q}};i_0,v_{i_0};x_1^1,\cdots,x_1^Q,x_2^1,\cdots,x_{t+1}^Q)$。

因此，消息 m 的最终签名 $\sigma=(\sigma_1,\cdots,\sigma_p)$。

T-ring-verify 算法：对于签名 σ_i，验证者计算

$$v_{k+1}=H(m,v_k\oplus G_i^k(x_k^1,\cdots,x_k^Q)),$$
$$k=i_0,\cdots,t+1,1,2,\cdots,i_0-2$$

其中 $v_{t+2}=v_1$，并验证 $v_{i_0}=H(m,v_{i_0-1}\oplus G_i^{i_0-1}(x_{i_0-1}^1,\cdots,x_{i_0-1}^Q))$ 是否成立。

若所有的 $\sigma_i(i=1,2,\cdots,p)$ 都能验证通过，则 σ 是消息 m 的有效签名。

在改进方案中，$v_{j+1}=H(m,H(s^1,\cdots,s^Q))$，如果某个 $s^k,k\in\{1,\cdots,Q\}$ 发生改变，则 $H(s^1,\cdots,s^Q)$ 会发生改变，需要更新所有的 $x_j^1,\cdots,x_j^Q$，使

$$v_{j+1}=H(m,v_j\oplus G_i^j(x_j^1,\cdots,x_j^Q))$$

因此，必须知道 G_i^j 的所有陷门，即需要所有签名者对单向函数求逆。而攻击者无法获知单向映射 G_i^j 的所有陷门，因此无法更新所有的 $x_j^1,\cdots,x_j^Q$ 生成

有效签名。

改进方案解决了原方案存在的问题，是一个安全的环签名方案。

3.3 改进 TRS 方案安全性分析

本节对改进 TRS 方案进行分析，包括签名的强不可伪造性及签名者的匿名性。

1) 强不可伪造性

在随机预言模型中，假设敌手最多可勾结($n-t-1$)个非签名者，在经过 q_H 次散列询问后，能以概率 ε 伪造($n-t,n$)门限环签名，则对于陷门单向映射 G_i^j，已知某个输出 y_0，可构造算法以概率 $\varepsilon' = \dfrac{1}{q_H(q_H-1)}\varepsilon$ 找到输入 x_0，使 $y_0 = G_i^j(x_0)$。

证明 本文需要在选择消息攻击下证明门限环签名的强不可伪造性（SU-TRS-CMA）。游戏包括 2 个参与者：挑战者和敌手。假设敌手在经过询问后可伪造有效的门限环签名，则挑战者已知陷门单向映射 G_i^j 的输出 y_0，可构造算法找到对应输入 x_0。具体步骤如下。

setup 挑战者为所有用户生成公私钥对，并将所有用户公钥 $PK_1,\cdots,PK_n$ 发送给敌手。

阶段 1 敌手适应性地进行下列询问。

散列询问：敌手发送字符串给挑战者，挑战者随机选择 Ql bit 的随机数 v，并设定某 2 个字符串的散列值分别为 v和$v\oplus y_0$，其余字符串选择 Ql bit 的随机数作为散列值，并返还给敌手。

私钥询问：敌手发送用户公钥给挑战者，挑战者返回对应私钥给敌手。敌手至多可询问($n-t-1$)个用户的私钥。

签名询问：敌手发送消息 m 给挑战者。挑战者随机选择($n-t$)个环成员私钥 $\mathrm{SK}_{i,1},\cdots,\mathrm{SK}_{i,n-t}$，根据 T-ring-sign 算法生成签名 σ，并返回给敌手。

挑战阶段 假设敌手能以概率 ε 伪造消息 m^* 的环签名 $\sigma^* = (\sigma_1^*,\cdots,\sigma_p^*)$，其中 $\sigma_i^* = (\mathrm{PK}_{p_i^{1,1}},\cdots, PK_{p_i^{1,Q}}, \mathrm{PK}_{p_i^{2,1}},\cdots,\mathrm{PK}_{p_i^{t+1,Q}}; i_0^*, v_{i_0}^*; x_1^{1,*},\cdots,x_1^{Q,*}, x_2^{1,*},\cdots,x_{t+1}^{Q,*})$，且$(m^*,\sigma^*)$没有出现在签名询问阶段。

如果 $G_i^j(x_j^{1,*},\cdots,x_j^{Q,*}) = y_0$，挑战者输出 $x_0 = x_j^{1,*}\|\cdots\|x_j^{Q,*}$，否则输出“错误”。

当且仅当 $v = H(s^1,\cdots,s^Q)$，$v_j = v\oplus y_0$ 时，$G_i^j(x_j^{1,*},\cdots,x_j^{Q,*}) = y_0$。由于敌手进行了 q_H 次散列询问，因此挑战者成功输出 x_0 的概率 $\varepsilon' = \dfrac{1}{q_H(q_H-1)}\varepsilon$。

2) 匿名性

设 $\pi_i^j = \{p_i^{j,1},\cdots,p_i^{j,q_i^j}\}$ 是($n,t+1$)–完备拆分系统中的合法子群。对于签名 σ_i，满足以下条件。

$$
\begin{aligned}
v_{j+1} &= H(m, v_j \oplus G_i^j(x_j^1,\cdots,x_j^Q))\\
&\vdots\\
v_1 &= H(m, v_{t+1} \oplus G_i^{t+1}(x_{t+1}^1,\cdots,x_{t+1}^Q))\\
&\vdots\\
v_j &= H(m, v_{j-1} \oplus G_i^{j-1}(x_{j-1}^1,\cdots,x_{j-1}^Q))
\end{aligned}
$$

其中，$x_k^1,\cdots,x_k^Q \in \{0,1\}^l, k = j+1,\cdots,t+1,1,2,\cdots,j-1$ 是随机选择的，只有 $x_j^1,\cdots,x_j^Q$ 是通过陷门单向函数 G_i^j 得到的。但对于攻击者来说，所有的 $x_k^1,\cdots,x_k^Q, k = 1,2,\cdots,t+1$ 均满足上述关系，无法区分是随机选择还是通过计算得到的，从而不能获得签名者的身份，即改进环签名方案满足匿名性。

3) 效率分析

假设系统中有 n 个用户，其中 $n-t$ 个用户可代表所有用户生成环签名。当 t 很小，即签名人数很多时，所提改进方案是效率最高的，如表 1 所示。为了公平起见，本文只比较传统的门限环签名方案。

在表 1 中，$t \ll n$，$n-t \approx n$，l 代表陷门单向映射的输出长度，g_i、g_i^{-1} 分别代表陷门单向映射及其逆映射。以 RSA（Rivest, Shamir, Adleman）加密算法为例，g_i、g_i^{-1} 相当于 RSA 的加密和解密运算，即模指数运算，q 代表双线性群的阶，“BP”代表“bilinear pairing（双线性对）”。通常，$q \geqslant 160$，

表 1 不同方案的门限环签名效率比较

方案	签名长度/bit	签名复杂度
BSS[8]	$O(l2^{n-t}n\mathrm{lb}n)$	$(n-t)g_i^{-1} + O(2^{n-t}n\mathrm{lb}n)g_i$
CWL[10]	$O(q^{n-t}n^2\mathrm{lb}n)$	$O(q^{n-t}n^2\mathrm{lb}n)BP$
I-TRS	$O(tl2^t n^2\mathrm{lb}n)$	$O(n2^t\mathrm{lb}n)g_i^{-1} + O(t2^t n^2\mathrm{lb}n)g_i$

而双线性对的计算代价近似于模指数运算。因此，所提方案在传统的门限环签名中是效率最高的。

4　可删除的区块链

本节基于改进的门限环签名方案，提出可删除的区块链结构，并做仿真实验进行模拟实现。当区块链数据过期时，只有绝大多数用户同意，并生成有效的门限环签名才能进行删除，否则不能进行删除。除删除操作外，不能对区块数据做其他更改。在删除区块数据时，不能影响其他区块数据的使用和验证。

4.1　数据删除概述

假设某个区块 i+1 的交易数据因为数据过期、废弃等原因，不再具有链上存储以供溯源的意义。继续存储该区块的交易数据无疑是对存储资源的浪费。在此情况下，网络中的相关节点向网络广播“删除区块 i+1 交易数据”的请求信息，具体表示为：DelTx={number, reason}，其中，number=i+1 为请求删除的交易数据所在的区块号，reason 即为请求删除的原因。其余所有合法节点在接收到 DelTx 后，会对该删除操作的合理性进行考证，比如该区块交易数据是否真的来自一个已经宣布倒闭的银行等等。考证过后，合法节点需广播自己关于 DelTx 的意见，记为 ReDelTx。ReDelTx=1，表示节点同意删除请求；ReDelTx=0，表示不同意删除请求。最终，每个节点均可统计整个网络对该删除信息的反馈，如果“同意”信息超过设定门限（本文设为全网节点的 75%），便认为该删除请求合法，然后生成一条专属的删除消息 m，其具体格式将在第 4.2 节中介绍。

接下来，对该删除消息持“同意”意见的节点将进行交易数据的删除工作，并成为环签名系统中的签名节点，按照第 3 节中提出的改进门限环签名方案，代表整个系统生成消息 m 的门限环签名，作为对删除消息 m 合法性的承诺。同时还将生成的环签名存储于交易数据原本所在的位置，并在全网进行广播，以供所有节点对该删除操作的合法性进行验证。

所提方案是在公链模式下，基于 PoSpace 共识机制提出的区块链数据删除方案。在目前使用 PoSpace 共识机制的区块链中，链上交易数据都是公开的，可供全网节点公开验证，不能实现交易数据的隐私性。因此，方案面向的区块数据都是公开可验证的，删除交易数据时对数据进行公开考证是合理的，删除操作不会增加数据的隐私泄露风险。如果用户对隐私性要求较高，可将数据存储在能实现隐私保护的区块链中，如 Zerocash[16]、Monero[17]等。

4.2　区块链结构

假设第 i+1 个区块数据可以删除，经大多数用户同意并生成门限环签名，对数据进行删除。可删除的区块链结构如图 3 所示。

在第 i+1 个区块中，包含 3 个子块：证明子块 φ_{i+1}（hash φ_{i+1}），签名子块 σ_{i+1}（signature σ_{i+1}）和门限环签名子块 TRS_{i+1}（T-ring-sig TRS_{i+1}）。证明子块和签名子块如第 2.3 节所述，这里只介绍门限环签名子块 TRS_{i+1}。

门限环签名子块 TRS_{i+1} 是多名用户集合 U 对消息 m 的签名，其中 m 包含：1) 当前区块序号 i+1；2) 删除相关信息 D_{i+1}，包括删除时间、原因等；3) 记账者对被删除交易子块 τ_{i+1} 的签名 ζ_τ。即 $\mathrm{TRS}_{i+1}=\mathrm{TRS}_U(i+1, D_{i+1}, \zeta_\tau)$，而从 TRS_{i+1} 不能得到集合 U 中签名用户的身份。

区块数据删除后，由签名节点广播至全网。其余用户首先验证门限环签名是否正确，然后将 TRS_{i+1} 中对被删除交易子块 τ_{i+1} 的签名 ζ_τ 与签名子块 σ_{i+1} 的签名 ζ_τ 进行比较，若一致则接受删除行为，否则不接受。

对比交易数据删除前后的区块信息可知，除交易数据被替换成相应环签名外，当前区块及其前后

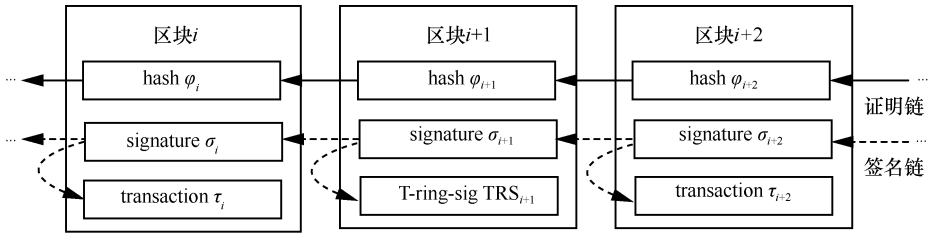

图 3　可删除的区块链结构

区块中的其余信息都未发生任何变化，区块链的结构未发生变动。可见，所提区块链方案可以成功删除区块中废弃的交易数据，释放大量存储空间，且不影响其他区块数据的使用和验证。

4.3 数据删除安全性分析

本文通过阈值设定，在全网 75%节点同意时，才可生成删除消息 m，并由同意删除的节点生成 m 的门限环签名，承诺数据删除的合法性。门限环签名本身具有的正确性、匿名性和强不可伪造性保证了该承诺的安全有效，并可供任意节点随时进行验证。其中，阈值比例 75%的设定是综合考虑“门限环签名效率”“节省存储空间”“删除操作是否代表大多数节点的意见”这 3 点，并通过实验测试得出的。一方面，阈值越高，越能代表系统中更多节点的意见，让删除操作更加安全。但另一方面，阈值过高将导致删除区块耗时太长，影响删除操作的效率，降低数据删除的应用价值。因此权衡两者，将阈值比例设定为 75%，可在保障安全性的前提下也保证方案效率。

同时，由第 4.2 节的分析可知，整个交易数据的删除过程并不会破坏原有区块链的结构。区块数据删除后，门限环签名被存储于原有数据所在位置。当前区块的证明子块和签名子块信息并未发生改变，因此该区块仍然与前后区块保持合法的链接关系。交易数据删除后，区块中签名子块关于交易数据的签名与门限环签名中包含的交易数据签名是一致的，其他网络节点将通过验证该一致性以确定区块链接的合法性。

综上，可删除的区块链具有以下特点。

1) 只有大多数用户同意才能对数据进行删除，避免恶意删除。由门限环签名的强不可伪造性可知，当区块数据过期或失效时，只有超过门限个数的用户才能生成有效的环签名，否则不能生成签名并通过验证。因此，本文可以设定合适的门限值，确保绝大多数用户同意才能删除区块数据，避免恶意用户的删除。

2) 除删除操作外，不能对区块数据做其他更改，例如插入数据、删除部分数据等。在基于空间证明的区块链中，每个区块中都包括证明子块和签名子块，证明子块可以认证记账者的身份，签名子块是记账者对交易数据的认证。如果恶意用户更改区块数据，例如插入数据、删除部分数据等，则需要重新生成签名子块，但是记账者的签名是不能伪造的，因此数据不会更改成功。

3) 第 i+1 个区块数据的删除并不影响其余区块的链接。原因如下：交易数据的删除只是把原有区块中的交易子块 τ_{i+1} 变成了门限环签名子块 TRS_{i+1}，并不改变证明子块 φ_{i+1} 和签名子块 σ_{i+1}，因此它们与前后区块的链接关系不受影响。

4.4 实验仿真

本节分别对区块的生成和删除进行了仿真实现。挖矿节点使用的实验设备配置为 2.4 GHz Intel i5 处理器，4 GB 内存，并在 Visual Studio Ultimate 开发环境下，使用 C++语言编程实现。在生成区块链时，散列函数和记账节点签名分别使用 SHA（secure hash algorithm）-256 和 DSA（digital signature algorithm）-512 实现。在删除区块链时，使用第 3 节提出的门限环签名，其中的单向陷门函数使用 RSA-1024 实现。

4.4.1 基于空间证明的记账权竞争

如第 2.3 节所述，在 PoSpace 共识机制下，矿工们基于空间证明对记账权展开竞争，首先模拟空间竞争过程。

假设系统生成如图 4 所示的有向图，共包含 20 个顶点。

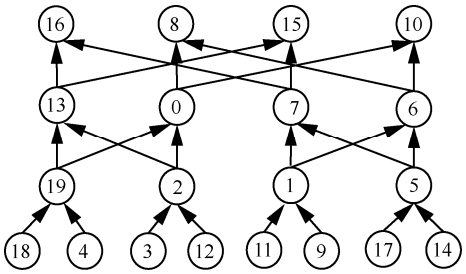

图 4 系统有向图

设有 5 位矿工，分别记为 A、B、C、D、E，每位矿工都可根据自己的公钥，计算该有向图中每个顶点的标签值（使用 SHA-256 实现），生成结构固定但顶点标签值与身份一一对应的专属有向图。由于空间限制，部分矿工无法一次性存储全部顶点，只能完成对部分顶点的存储，其他非存储顶点则需要依据有向图结构和已存储的顶点计算得到。当系统随机要求矿工返回相关顶点标签值时，本地空间大、存储顶点多的矿工总是拥有较大的优势，以最快的速度响应系统，从而赢得记账权，这样便形成了基于空间的记账权竞争。其中，5 位矿工的公钥及其对图 4 有向图的存储情况如下。

PK_A：CA22D0C6，至多存储 8 个顶点。

PK_B：F3EDCC34，至多存储 11 个顶点。

PK_C：0650EE12，至多存储 14 个顶点。

PK_D：87A30060，至多存储 17 个顶点。

PK_E：72DC0E20，至多存储 20 个顶点（全图存储）。

每个矿工都作为一个空间证明者 P 向系统扮演的验证者 V 证明自己拥有足够的空间存储全图，V 多次发布验证顶点标号集合，记为 $\mathrm{Ch}(c_1,c_2,\cdots,c_k)$，向 P 发起挑战，要求 P 返回 Ch 中每个标号对应的顶点标签值及其母顶点标签值，以验证 P 是否有足够大的空间存储全图，P 返回相应数值响应 V 发起的挑战。表 2 记录了上述 5 位矿工在不同挑战 Ch 下的空间证明时间。

表 2　　5 名矿工空间证明时间比较

矿工	Ch={13,10,15,8,16}	Ch={1,5,6,7,0,13,10,15,8,16}	Ch={18,4,3,19,2,1,5,6,7,0,13,10,15,8,16}	Ch={12,11,9,17,14,18,4,3,19,2,1,5,6,7,0,13,10,15,8,16}
A	410 ms	480 ms	526 ms	548 ms
B	325 ms	386 ms	431 ms	456 ms
C	188 ms	254 ms	297 ms	325 ms
D	121 ms	185 ms	229 ms	267 ms
E	65 ms	129 ms	170 ms	193 ms

由表 2 可知，在不同的挑战 Ch 下，P 的空间越大，对 V 的挑战的响应就越快，越容易获得记账权，这便是基于空间证明的 PoSpace 共识机制下区块记账权竞争的基本过程。

4.4.2　新区块的生成

当节点获得记账权后，依次生成新区块的证明子块（hash 子块）、签名子块和交易子块。原始区块 33~35 的数据如图 5 所示，本文以区块 35 为例进行说明。

1) 证明子块，即 $\varphi_i=\mathrm{hash}(i,\zeta_\varphi,(p_{k_i},\gamma_i,\ c_i,a_i))$。其中，$i$=35；$\zeta_\varphi$ =786B38596DB4D9B772E732B7453BBB4122C7AF6F，即记账者对区块 34 的签名子块 φ_{34} 进行 DSA 签名所得；$(p_{k_i},\gamma_i,c_i,a_i)$ 为空间证明中的证明信息，p_{k_i} 为记账者公钥，p_{k_i} = 0650EE12，γ_i 为记账者使用 Merkle Tree 对系统有向图结构的证明，γ_i =d9350ef7422d90eedda82bdde90cc2d4ab3defe513ec0ded960e59674b5edf9a，c_i 为记账者在空间证明过程中收到的来自验证者的挑战，即验证顶点的标号，c_i=(0,7,8,16,13)，a_i 是记账者对 c_i 的回应，即 c_i 中的顶点的标签值及其母顶点的标签值。对以上数据进行 SHA-256 散列计算，即可得到区块 35 的证明子块结果，即

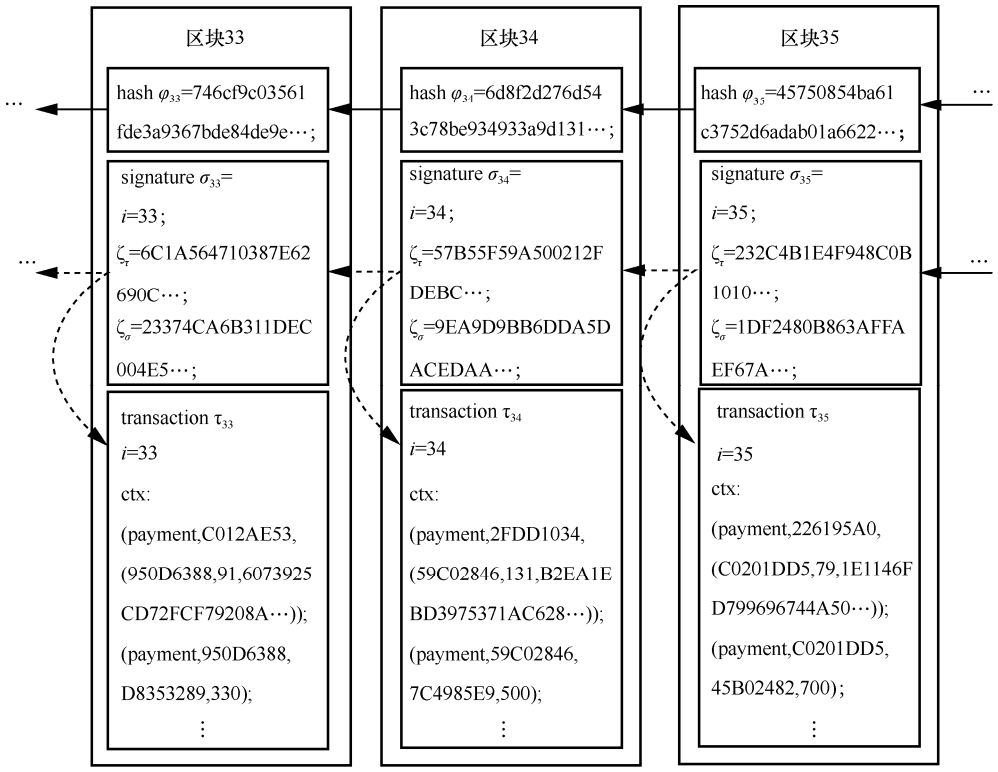

图 5　原始区块 33~35 的数据

φ_{35} =45750854ba61c3752d6adab01a6622876daa8d3749d128167ac86fdd27cd4fbf 。

2) 签名子块，即 $\sigma_i=\{i,\zeta_\tau,\zeta_\sigma\}$。其中，$i$=35；$\zeta_\tau$ 为记账者对区块 35 的交易子块 τ_{35} 进行 DSA 签名所得，ζ_τ =232C4B1E4F948C0B1010DB2B8A3FB5C618AC4F04；ζ_σ 则为记账者对前一区块，即区块 34 的签名子块进行 DSA 签名所得。以上 3 个部分便构成了区块 35 的签名子块。

3) 交易子块，即 $\tau_i=\{i,\text{ctx}\}$。其中，i=35，ctx 包含 50 条交易信息。

4.4.3 区块的删除

假设当前区块链中，编号 33~35 的区块信息如图 5 所示。其中，区块 34 中的交易数据已过期，为了节约存储空间，区块链中的各个节点于 2018 年 7 月 3 日达成共识，对该区块执行删除操作。首先，由区块 34 删除时间“20180703”、删除原因“overdue”和记账者对区块 34 中交易数据的签名“57B55F59A500212FDEBC6F92BC8B14F46444BFDE”，生成删除行为对应的消息 m=“3420180703overdue57B55F59A500212FDEBC6F92BC8B14F46444BFDE”，并按照第 3.2 节中的改进算法，对消息 m 生成门限环签名，记录在原始交易子块中。

假设系统中总用户数 n=8，环签名门限值为 6，则非签名用户数 t=2。设签名用户集合记为 $\{P_1,P_2,P_3,P_4,P_5,P_6\}$，非签名用户集合记为 $\{P_7,P_8\}$。设某次拆分方法为：$\pi_i=\{\pi_i^1,\pi_i^2,\pi_i^3\}$，$\pi_i^1=\{P_1,P_2,P_3,P_4,P_5,P_6\}$，$\pi_i^2=\{P_7\}\}$，$\pi_i^3=\{P_8\}$，得到一个公平拆分，其中，$\pi_i^1$ 是合法子群，且 Q=6。门限环签名由 π_i^1 中的所有用户代表整个系统对消息 m 进行签名，其中单向陷门映射由 RSA 算法实现，对应的陷门便是解密算法中的私钥。本文首先选取 1 024 bit 的公共参数，即 N=BE192C23548573ACA5DB62B 0507123210517D14D6E4D19CE0931BD63A212DB51E855FDFF92D461C9AFD62DBB6A825DCEB3030452ECE1890D1050DCBDA4A8B5FE1462894AAD22825C04DF4B164A49C0183466E5DC34C9B1BB29E414C4F73AAC9508E9C1A6423D3BE7E5C067D656BE928024619B2335EE29CBB39B81478BA512A1，然后随机选取 $(t+1)\times Q=3\times 6=18$ 对公私钥 (e_i,d_i)，以此构成不同的单向陷门映射，完成环签名操作。最终对消息 m 生成环签名

$$\sigma_i=(PK_{p^{1,1}},\cdots,PK_{p^{1,6}},PK_{p^{2,1}},\cdots,PK_{p^{3,6}};$$
$$i_0,v_{i_0};x_1^1,\cdots,x_1^6,x_2^1,\cdots,x_3^6)$$

其中，$x_2^1,\cdots,x_2^6$，$x_3^1,\cdots,x_3^6$ 为 2 组随机数，作为前 2 次单向陷门映射的输入，而 $x_1^1,\cdots,x_1^6$ 只能由签名者使用陷门，即 RSA 算法中的解密私钥才能得到。求解 x_1^1 程序运行如图 6 所示，$x_1^2,\cdots,x_1^6$ 求解过程与之类似，故在本文省略。

将上述过程重复 p 次，得到最终的门限环签名 $\sigma=(\sigma_1,\cdots,\sigma_p)$，其中，$p=2^{2+1}1\text{b}8=24$。

当区块 34 中的交易数据删除后，区块 33~35 的数据如图 7 所示。

删除操作完成并广播后，网络中所有节点均能对门限环签名进行验证，并将消息 m 中记账者对区块 34 中交易数据的签名与签名子块中的签名进行对比验证，如果验证通过，则认可数据删除行为的合法性。

最后，本文对生成和删除区块的效率进行了分析。如第 4.3 节所述，本文需要选定合适的阈值，以权衡“门限环签名效率”“节省用户存储空间”以及“删除操作是否代表大多数节点的意见”这 3 点，使其在保障安全性的前提下也保证方案效率，其中，效率分析包括计算时间与存储空间比较，即删除操作用时越短，环签名所占空间越小，方案效率越高。在以下实验中，本文选取不同阈值，从“删除区块时间”和“删除后节省空间占比”这 2 个方面进行了比较。

如第 3.1 节所述，对于所有的整数 n 和 t，且 $t<n$，$(n,t+1)$–完备拆分系统一定是存在的。同时，为了使

图 6　求解 x_1^1 程序运行

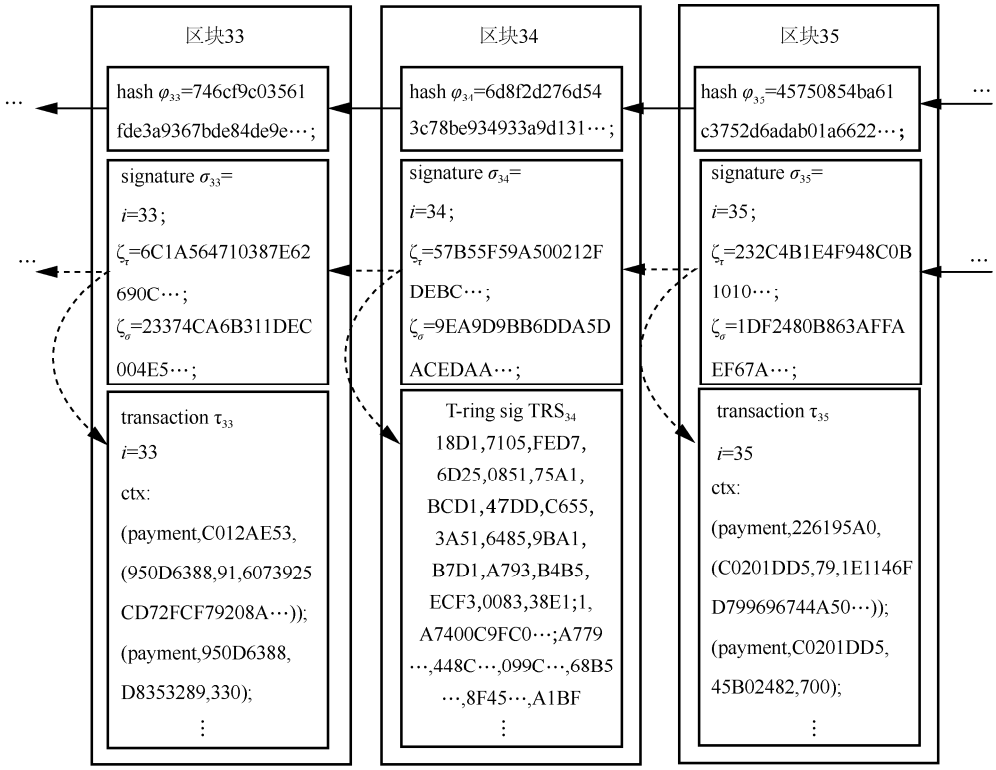

图 7　区块 34 交易数据删除后区块 33~35 的数据

删除操作代表系统多数节点意见以保证安全，签名节点应在半数以上。以上实验中 n=8，因此 t={1,2,3}，合法阈值集合 $n-t$={5,6,7}。表 3 和表 4 分别比较了不同阈值下区块删除的平均时间和删除前后的存储空间及节省空间占比。

表 3　不同阈值下区块删除的平均时间比较

阈值	区块删除的平均时间/s	区块生成的平均时间/s
5	2.51	
6	3.03	0.965
7	3.48	

表 4　不同阈值下交易删除前后的存储空间及节省空间占比

阈值	数据删除后/KB	原有区块/KB	节省空间占比
5	153.77		61.55%
6	73.7	400	81.58%
7	32.16		91.96%

由表 3 可知，区块删除的平均时间随着阈值的增大而增加，三者之间两两相差大约 0.5 s，约占区块生成平均时间的$\frac{1}{2}$，对计算效率影响明显。由表 4 可知，当阈值为 6 和 7 时，交易删除后节省空间的效果显著，均大于 80%；而阈值为 5 时，交易删除后节省空间大约 60%。因此，当阈值为 7 时，虽然删除后节省空间的效果最优，但删除区块耗时较长；当阈值为 5 时，虽然删除区块耗时较短，但节省空间效果不太理想，且参与删除的节点相对较少，降低了方案的安全性。综合考虑区块删除的安全性和效率，本文选择阈值为 6，即占比$\frac{6}{8}$=75%是同时保证方案安全性和效率的最佳阈值。

如表 5 所示，当阈值比例为 75%时，生成一个区块平均耗时 0.965 s，删除交易数据并生成门限环签名平均耗时 3.028 s。

表 5　5 个区块在阈值比例为 75%时的生成区块和删除区块耗时

区块	生成区块耗时/s	删除区块耗时/s
区块 1	1.112	3.044
区块 2	0.873	3.027
区块 3	0.897	3.041
区块 4	0.936	3.017
区块 5	1.008	3.013
平均	0.965	3.028

5 结束语

本文首先提出了改进的门限环签名方案，在随机预言模型中可证安全，同时满足签名的强不可伪造性与签名者的匿名性。然后，基于空间证明的区块链结构，使用门限环签名方案提出了可删除的区块链。当某个区块数据过期失去存储价值时，经大多数节点同意后可有效删除该区块，大大节省了用户的存储空间，并保持区块链的总体结构不变。实验结果表明，所提区块链方案生成和删除区块的效率都很高，且不影响其他区块的存储和使用。

参考文献：

[1] NAKAMOTO S. Bitcoin: a peer-to-peer electronic cash system[R]. Bitcoin, 2009.

[2] 袁勇, 王飞跃. 区块链技术发展现状与展望[J]. 自动化学报, 2016, 42(4): 481-494.
YUAN Y, WANG F Y. Blockchain: the state of the art and future trends[J]. Acta Automatica Sinica, 2016, 42(4): 481-494.

[3] GARAY J, KIAYIAS A, LEONARDOS N, et al. The bitcoin backbone protocol: analysis and applications[C]//Annual International Conference on the Theory and Applications of Cryptographic Techniques. 2015: 281-310.

[4] LARIMER D. Transactions as proof-of-stake[R]. White Paper, 2013.

[5] AGGELOS K, ALEXANDER R, BERNARDO D, et al. Ouroboros: a provably secure proof-of-stake blockchain protocol[C]//Annual International Cryptology Conference. 2017: 357–388.

[6] PARK S, KWON A, FUCHSBAUER G. SpaceMint: a cryptocurrency based on proofs of space[R]. Financial Crypto, 2018.

[7] RIVEST R, SHAMIR A, TAUMAN Y. How to leak a secret[C]// International Conference on the Theory and Application of Cryptology and Information Security. 2001: 552-565.

[8] BRESSON E, STEM J, SZYDLO M. Threshold ring signatures and applications to ad-hoc groups[C]//Annual International Cryptology Conference. 2002: 465-480.

[9] TOSHIYUKI I, KEISUKE T. An (*n*-*t*)-out-of-*n* threshold ring signature scheme[C]//Australasian Conference on Information Security and Privacy. 2005: 406-416.

[10] CHUNG Y, WU Z, LAI F, et al. A novel ID-based threshold ring signature scheme competent for anonymity and anti-forgery[C]//International Conference on Computational and Information Science. 2007: 502-512.

[11] MELCHOR C, CAYREL P, GABORIT P, et al. A new efficient threshold ring signature scheme based on coding theory[J]. IEEE Transactions on Information Theory, 2011, 57(7): 4833-4842.

[12] ZHANG J, ZHAO Y. A new multivariate based threshold ring signature scheme[C]//International Conference on Network and System Security. 2015: 526-533.

[13] KRAWCZYK H, RABIN T. Chameleon Hashing and signatures: US Patent 6108783 [P]. 2000-08-22.

[14] LI P, XU H, MA T. Research on fault-correcting blockchain technology[J]. Journal of Cryptologic Research, 2018, 5(5): 501-509.

[15] BONEH D, SHEN E, WATERS B. Strongly unforgeable signatures based on computational Diffie-Hellman[C]//International Workshop on Public Key Cryptography. 2006: 229-240.

[16] SHI F S, MAN H A, JOSEPH K. RingCT 2.0: a compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency monero[C]//European Symposium on Research in Computer Security. 2017: 456-474.

[17] SASSON E, CHIESA A, GARMAN C, et al. Zerocash: decentralized anonymous payments from bitcoin[C]//IEEE Symposium on Security and Privacy. 2014: 459-474.

[作者简介]

任艳丽（1982− ），女，山西运城人，博士，上海大学教授、博士生导师，主要研究方向为公钥密码学、可验证外包计算、区块链安全等。

徐丹婷（1994− ），女，浙江绍兴人，上海大学硕士生，主要研究方向为密码学与区块链。

张新鹏（1975− ），男，黑龙江鸡西人，博士，上海大学教授、博士生导师，主要研究方向为多媒体信息安全、信息隐藏、数字取证、图像处理等。

谷大武（1970− ），男，河南漯河人，博士，上海交通大学教授、博士生导师，主要研究方向为密码分析与设计、信息分析与密码工程、计算机安全体系结构等。