基于动态时间型二叉树的隐蔽通信模型

佘维[1,2]，马佳伟[1,2]，张淑慧[1,2]，程孔[1,2]，刘炜[1,2]，田钊[1,2]
（1.郑州大学网络空间安全学院，河南 郑州 450002；2.郑州市区块链与数据智能重点实验室，河南 郑州 450002）

摘　要：针对区块链隐蔽通信效率与安全问题，提出一种基于动态时间型二叉树的隐蔽通信模型。通过特定时刻动态生成时间型二叉树，利用根哈希提取随机因子为树节点分配不同的路径编码，将通信信息字符映射为编码路径及索引字段。通过将特定时刻嵌入路径编码空置位并整合至区块链交易实现隐蔽传输，接收端通过解析特定时刻重构时间型二叉树完成解码。实验结果表明，相较于同类模型，所提模型在保证安全性的同时提升了通信效率，并避免了预协商过程带来的安全隐患。

关键词：区块链；动态时间型二叉树；路径编码；隐蔽通信
中图分类号：TP309.2
文献标志码：A
DOI：10.11959/j.issn.1000–436x.2025031

Covert communication model based on dynamic time binary trees

SHE Wei[1,2], MA Jiawei[1,2], ZHANG Shuhui[1,2], CHENG Kong[1,2], LIU Wei[1,2], TIAN Zhao[1,2]
1. School of Cyber Science and Engineering, Zhengzhou University, Zhengzhou 450002, China
2. Zhengzhou Key Laboratory of Blockchain and Data Intelligence, Zhengzhou 450002, China

Abstract: A covert communication model based on a dynamic time binary tree was proposed to address the efficiency and security issues in blockchain-based covert communication. A time binary tree was dynamically generated at specific moments, and the roothash was used to extract random factors for assigning different path codes to the tree nodes. Communication information characters were mapped to the encoded paths and index fields. By embedding vacant bits of the path codes at specific moments and integrating them into blockchain transactions, covert transmission was achieved. The receiver reconstructed the time binary tree at the specific moment to complete the decoding. Experimental results show that, compared to similar models, the proposed model improves communication efficiency while ensuring security, and avoids the security risks brought by the pre-negotiation process.

Keywords: blockchain, dynamic time binary tree, path encoding, covert communication

0　引言

近年来，秘密消息传输遭遇拦截并被破解的事件频发，消息传输的隐蔽性和保密性以及载密消息的抗检测性日益受到重视[1]。传统通信方式依赖于直接的明文传输或经过简单的加密过程形成密文传输，然而，这些方法往往呈现出明显的特征，使得攻击者能够识别、截取并尝试破解信息，或者利用重放攻击来损害消息的正确性和完整性。因此，开

收稿日期：2024–12–24；修回日期：2025–02–11
通信作者：田钊，tianzhao@zzu.edu.cn
基金项目：国家重点研发计划基金资助项目（No.31703-3）；河南省高等学校重点科研基金资助项目（No.24A520045）
Foundation Items：The National Key Research and Development Program of China (No.31703-3), The Key Scientific Research Project of Colleges and Universities in Henan Province (No.24A520045)

发出使消息难以被攻击者察觉的通信技术是解决上述问题的关键。

隐蔽通信是一种将秘密消息嵌入正常通信过程中的传输技术，能够有效地减少秘密消息被检测的可能性，从而降低秘密信息暴露的风险[2-4]。隐蔽通信采用公共传输媒介，与传统通信方式不同的是，隐蔽通信将秘密信息隐藏在正常发送的信息中，而不是直接将原文或密文作为正常信息进行发送。只有特定的通信方知道如何识别隐蔽信息，而攻击者则需要耗费大量的计算资源才能对可能存在的隐蔽通信行为做出判断。

区块链作为一个去中心化的平台，为用户提供了一个可以隐藏真实身份的机会。在以比特币和以太坊为代表的公链中，由于网络节点间广播机制的存在，用户发送的信息即使不直接发送到接收地址，接收方也能间接接收信息。广播机制使用户之间不需要直接通信就可以实现消息的传输，因此在公链中的隐蔽通信行为即使被检测也无法追溯到现实中的用户，从而达到保护隐私的目的[5]。

最早的区块链隐蔽通信信道由Partala[6]提出，通过改变交易地址的最低位实现信息的隐藏。Tian等[7]基于动态标签构建隐蔽信道，用秘密消息替代椭圆曲线数字签名算法（ECDSA, elliptic curve digital signature algorithm）的私钥，并基于新私钥生成公钥pk，能有效地抵御统计分析和关联攻击并防止交易追溯。佘维等[8]提出基于马尔可夫链生成文本的隐蔽通信技术，通过维护同一转移概率矩阵并生成符合自然语言特征和高可读性的载密语句来隐藏秘密信息，再使用环签名技术保证匿名性，有效解决了区块链通信中隐蔽性不足和信息交叉等问题。黄冬艳等[3]基于区块链中的时间戳提出时间型隐蔽通信方案，通过发送交易的时间长短进行编码。不同于通常的存储型隐蔽通信，时间型隐蔽通信可以提高抗检测的安全性表现但是嵌入效率也会相应降低。佘维等[9]使用生成式对抗网络（GAN, generative adversarial network）生成含有载密信息的图像，结合密文策略属性基加密（CP-ABE, ciphertext policy attribute based encryption）和环签名技术使发送者匿名提高安全性，将信息存入链下星际文件系统（IPFS, inter planetary file system），确保了通信的安全性以及高传输效率。随着区块链在隐蔽通信领域的不断发展，仍然存在以下亟须解决的问题。

1) 使用传统编码方法保证信息安全性时会降低编码和解码效率，当传输信息较大时，消耗时间较长；此外，使用预协商交易的方式传递密钥信息、开始标识符及结束标识符信息和分片信息进一步降低了通信效率。

2) 基于区块链网络的信息编码方案大多采用固定映射的方式进行编码，更容易被破解；并且区块链网络中的信息是永久保存的，如果字符-编码的映射规则被破解，则可以追溯并破译之前所有发送的信息，这将大大降低加密信息的安全性；此外，每次通信前发送预协商交易的方式会破坏原有发送规律导致通信的隐蔽性削弱，进一步降低通信的安全性。

为了解决上述问题，本文提出了基于动态时间型二叉树的隐蔽通信模型。通信方之间各自维护相同的时间型二叉树，将字符信息存入树中每个节点的数据字段中，使用随时间改变的根哈希作为随机因子载体，不断改变每个节点的路径编码。发送方将信息中的每个字符转换为路径编码并生成相应的索引字节后，在路径编码的空置位中嵌入选择的时刻，将嵌入后的路径编码字段和索引节点字段拼接后嵌入交易并发送到区块链网络。接收方从网络获取交易，提取发送方嵌入的时刻并恢复信息发送时的树形及根哈希，根据路径编码和索引节点在本地时间型二叉树中查找相应字符，恢复原消息。

本文的主要贡献如下。

1) 利用了索引节点最大程度减少树高，统计了字符出现的概率，根据概率大小分配字符所在节点位置，提高编码和解码效率；此外，本文设计了关键信息嵌入路径编码空置位的方法，在每次通信前不需要发送预协商交易，进一步提高通信效率，解决了传统编码方法在保证信息安全性时会降低通信效率的问题。

2) 提出了基于动态时间型二叉树的隐蔽通信模型，字符-路径编码映射可以根据时间的不同而改变，提高信息传输的安全性；取消预协商交易过程，减少了隐蔽通信暴露概率以及关键信息泄露的风险，进一步提高通信的隐蔽性和安全性，解决了传统隐蔽通信方法编码固定和使用预协商交易导致安全性不足的问题。

1　相关技术

1.1　区块链隐蔽通信

区块链是一种按时序将数据区块以链式结构进行存储[10]，并以密码学方式保证不可篡改和不可伪造的去中心化共享账本[11]，具有去中心化、匿名性、可追溯、不可篡改和不可伪造等特性。区块链中的去中心化是指网络架构和数据管理不依赖于单一的中心机构，网络中的每个节点都保存有完整或部分的区块链数据副本[12]。设区块链为有序序列 $C=\{B_0,B_1,\cdots,B_n\}$，每个区块满足

$$B_i=\left\langle H_{i-1},T_i,\text{Nonce}_i,\text{RootHash}_i\right\rangle ,i\geqslant 1 \tag{1}$$

其中，$H_i=\mathcal{H}\left(B_i\right)$为区块哈希函数，满足抗碰撞性 $\Pr\left[\mathcal{H}(x)=\mathcal{H}\left(y\right)\right]\leqslant\varepsilon\left(x\neq y\right)$；$T_i$ 为交易数据集，通过 Merkle 树构建根哈希 RootHash_i；链式验证条件为 $\forall i>0,H_{i-1}=\text{PrevHash}\left(B_i\right)$。

传统隐蔽通信使用第三方通信媒介进行信息传输，受到系统中心化的现实约束，降低了通信的隐蔽性。同时传统隐蔽通信目标固定且单一，通信的安全性也受到影响。区块链拥有去中心化和匿名性等特性，信息通过洪泛法向多个邻节点传输，弥补了传统隐蔽通信借助中心化媒介和通信目标固定等不足。此外，区块链的不可篡改和不可伪造等特性也一定程度上保证了隐蔽通信的可靠性。

1.2　根哈希

根哈希的概念来自 Merkle[13]提出的 Merkle 树，是一种基于哈希值构建的二叉树，在非叶子节点中存放所有子节点的哈希值，在叶子节点中存放数据块的哈希值[14]。在区块链中，区块中的所有交易哈希值构成一棵 Merkle 树，区块头中包含 Merkle 树的根哈希值字段。所有用户都可以使用根哈希字段，快速验证区块内交易数据的完整性，防止交易数据被篡改从而保证交易的安全性。

假设交易集为 $T=\left\{\text{tx}_1,\text{tx}_2,\cdots,\text{tx}_m\right\}$，构造 Merkle 树

$$\begin{cases}H_i^{(0)}=\mathcal{H}(\text{tx}_i),\ 1\leqslant i\leqslant m & \text{叶子层}\\ H_j^{(k)}=\mathcal{H}(H_{2j-1}^{(k-1)}\|H_{2j}^{(k-1)}),\ 1\leqslant j\leqslant\dfrac{m}{2^k} & \text{非叶子层}\end{cases} \tag{2}$$

叶子层（即第 $k=0$ 层）直接对交易数据进行哈希；对于非叶子层的哈希节点，则进行递归运算，直到得到根哈希。其中，k 表示树的层数，$h_j^{(k)}$ 表示第 k 层的第 j 个哈希节点。

根哈希通常存放在 Merkle 树的根节点中，即 $\text{RootHash}=H_1^{(\lceil \text{lb}^m\rceil)}$。更改树中任意节点中的哈希值或节点数量均会导致根哈希的改变，可将其应用于动态变化的树形中保证安全性。

2　动态时间型二叉树说明

2.1　动态时间型二叉树定义

二叉树是指树中节点的度不大于 2 的有序树，时间型二叉树是在二叉树的基础上，增加了随时间变化而不断生成新的树叶子节点机制、节点存放信息机制和根哈希机制。

1) 从一个起始时间开始，此时时间型二叉树只有一个根节点或一个基础树形，每过一个时间间隔，树会生成一个新的叶子节点，即树的节点数量和深度会随时间不断变化。

2) 在原始的二叉树节点数据结构中增加数据（data）字段和在非叶子节点的数据结构中增加哈希字段，树中的每个节点都可以存储任意的数据信息包括空信息，非叶子节点可以额外存储其左右子节点（非叶子节点）的哈希值拼合之后的哈希值，并且节点中信息的改变不会导致树形的改变。

3) 在时间型二叉树的根节点中固定存储根哈希值，树中任意节点中的信息或节点数量的变化均会导致根哈希的改变。

2.2　生成树规则

时间型二叉树中，深度小于或等于 4 的节点中的 data 字段存放字符；深度大于 4 的节点不存放任何数据且随时间的增加不断生成新的节点，这类节点仅用于改变树的根哈希值，根哈希生成过程如图 1 所示。

从右子树的叶子节点开始，将左右叶子节点的数据部分分别取哈希得到 $\mathcal{H}\left(\text{data}_1\right)$ 和 $\mathcal{H}\left(\text{data}_2\right)$，在父节点 A 中存放节点 A 的哈希值 $\mathcal{H}(\text{A})=\mathcal{H}\left(\mathcal{H}\left(\text{data}_1\right)\|\mathcal{H}\left(\text{data}_2\right)\right)$，同理在左子树父节点 B 哈希字段中存放哈希值 $\mathcal{H}(\text{B})=\mathcal{H}\left(\mathcal{H}\left(\text{data}_3\right)\|\mathcal{H}\left(\text{data}_4\right)\right)$，在节点 C 中存放哈希值 $\mathcal{H}(\text{C})=\mathcal{H}\left(\mathcal{H}(\text{A})\|\mathcal{H}(\text{B})\right)$。递归过程结束得到 $\text{RootHash}=\mathcal{H}\left(\mathcal{H}(\text{C})\|\mathcal{H}(\text{D})\right)$。字符所在树节点的深度取决于字符出现概率的大小，定义字符集合 $C=\left\{c_1,c_2,\cdots,c_n\right\}$，其中每个字符 c_i 的出现

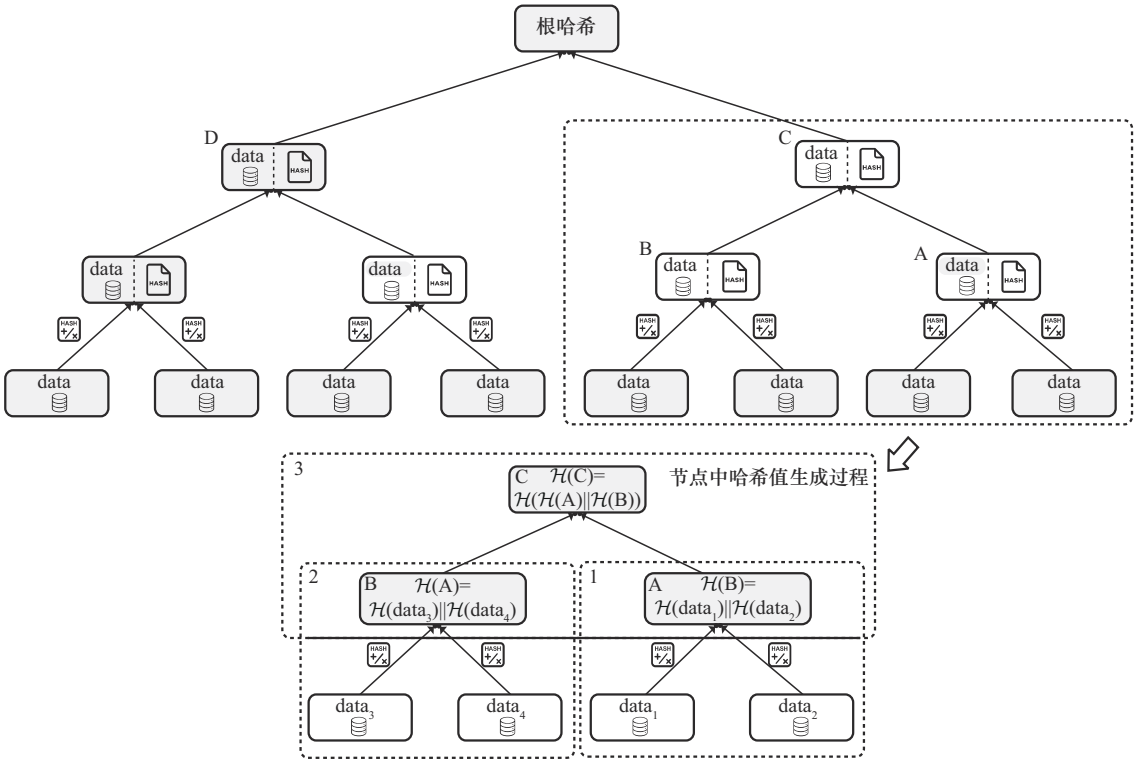


图 1　根哈希生成过程

概率为 $P(c_i)$。本文对 *Jane Eyre* 约 1 000 000 个字符、*Pride And prejudice* 约 700 000 个字符和 *Oliver Twist* 约 880 000 个字符进行字母类字符出现概率的统计，结果如表 1 所示。

表 1　　**字符出现概率统计**

字符	出现概率	字符	出现概率
e	0.102 3	c	0.019 2
t	0.068 4	y	0.018 5
a	0.061 5	f	0.017 7
o	0.059 3	w	0.017 4
n	0.055 9	g	0.014 7
i	0.052 5	p	0.012 0
h	0.049 7	b	0.011 9
s	0.048 6	v	0.008 5
r	0.047 9	k	0.004 6
d	0.032 4	z	0.001 4
l	0.031 1	x	0.001 2
u	0.022 3	q	0.000 9
m	0.019 5	j	0.000 8

设 D 为树的深度，节点 N 包含字符集合 C_N，$P_{\text{high}}(c)$ 和 $P_{\text{low}}(c)$ 为高频和低频字符的概率，构建平衡字符频率的时间型二叉树。

1) 字符集合 C 中元素按出现概率大小排序 $P(c_1) \geqslant P(c_2) \geqslant \cdots \geqslant P(c_n)$。

2) 将 $C_{1\sim4}$ 字符暂分布到 $D=1$ 的 2 个节点中，$C_{5\sim12}$ 字符暂分布到 $D=2$ 的 4 个节点中，$C_{13\sim28}$ 字符暂分布到 $D=3$ 的 8 个节点中，其余字符分布到 $D=4$ 的 16 个节点中。

3) 将 D 相同的字符按出现频率分为高-低频字符对放入一个节点中，例如在 $D=1$ 的字符 e、t、a、o，频率最高和最低的字符放入左子树第一个节点，$N_{\text{left}}=\left\{c_{\text{high}_1},c_{\text{low}_1}\right\}=\{\text{'e','o'}\}$，频率次高和次低的字符放入右子树另一个节点 $N_{\text{right}}=\left\{c_{\text{high}_2},c_{\text{low}_2}\right\}=\{\text{'a','t'}\}$，继续对 $1<D<4$ 的字符进行相同操作，直到所有节点中均包含一对高-低频字符对。通过平衡节点中包含字符的频率，可以避免某些节点的路径编码出现频率过高的问题。

4) 对于在 $D=4$ 的节点，使其包含 4 个字符而非 2 个字符，可以更好地减少遍历深度，并与后文提出的索引节点结构契合，从而提高模型编码与解码效率。

5) 时间型二叉树平衡频率后的最终结构如图 2 所示。$D \leqslant 4$ 的节点构成基础树形结构，不随时间生成；$D > 4$ 的节点从 00:00:00 开始随着时间的推移不断生成，仅用于改变根哈希。

2.3　树维护规则

通信方 $\{u_1,u_2,\cdots,u_i,\cdots,u_n\}$ 使用相同的生成树规则时，每个通信方每日在本地维护的时间型二叉树的树形数量是有限且相同的，若时间间隔 t 为秒级，则每个通信方每日在本地维护的树形数量为 $\dfrac{24\times 60\times 60}{t}$ 个。记每个通信方每日维护的树形数量集合为 σ，则 $\sigma_0=\sigma_1=\cdots=\sigma_i=\cdots=\sigma_n$。

发送方 u_s 选择某个时刻并发送，该时刻在 σ_s 中存在唯一对应的树形。接收方 u_r 获取到发送方选择的时刻后，在 σ_r 中也一定可以得到相应的树形，其中 $u_s,u_r\in\{u_1,u_2,\cdots,u_i,\cdots,u_n\}$。

后文所提到的根据时刻恢复树形，本质上是接收方 u_r 从集合 σ_r 中获取该时刻对应的树形。

3　动态时间型二叉树通信模型

本节提出一种基于动态时间型二叉树的隐蔽通信模型，通信方之间通过时间型二叉树的生成规则，各自在本地维护一棵时间型二叉树，发送方选取任意时刻的根哈希并提取随机因子改变节点路径。对原文进行路径编码后发送交易信息，接收方获取交易信息后根据嵌入的时刻还原树形，执行逆操作解码信息后得到原文，模型流程如图 3 所示。

本节将从信息编码、信息嵌入和信息提取对本文的基于动态时间型二叉树的隐蔽通信模型进行说明。

3.1　信息编码

发送方首先获取当前时刻下时间型二叉树的根哈希 RootHash，从中提取所有数字作为随机因子 seed 并记录，若 RootHash 中数字位数不足 5 位或

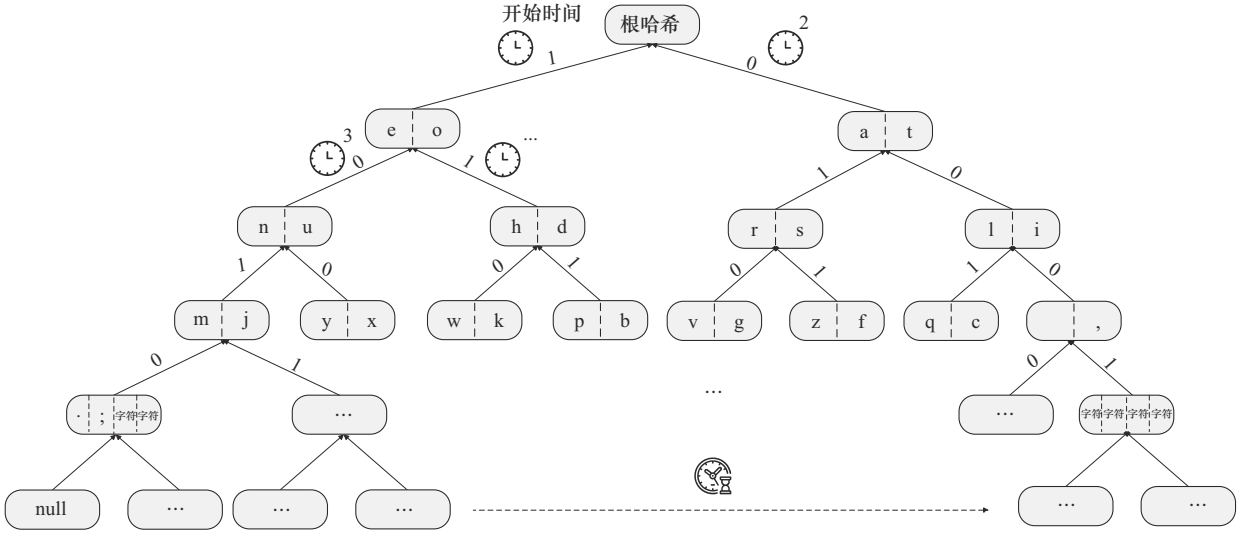


图 2　时间型二叉树平衡频率后的最终结构

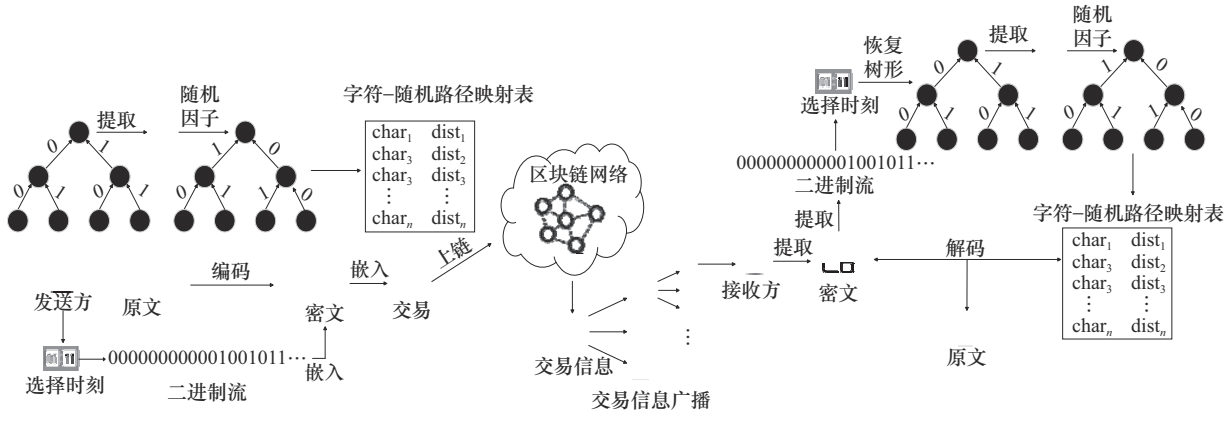


图 3　模型流程

seed与之前记录有碰撞，则对根哈希继续进行哈希运算，直至数字位大于5位或seed是唯一使用的。以十进制保存当前时刻或发送方选择的编码时刻selected_moment，其编码时刻二进制表示如图4所示。

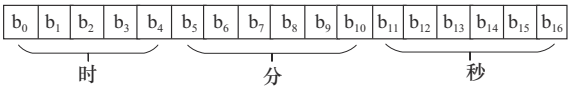


图 4 编码时刻二进制表示

时、分、秒分别占高5位、中6位和低6位，以尽可能短的长度表示完整时刻，减少路径编码空置位的占用。

应用随机因子改变树中每个分支的编码，从而改变每个节点对应的路径，结果如图5所示，具体字符-路径映射关系如表2所示。

表2 字符-路径映射关系

字符	原始路径编码	应用随机因子后的路径编码
e/o	0	1
a/t	1	0
n/u	00	10
h/d	01	11
r/s	10	01
l/i	11	00
⋮	⋮	⋮

路径编码随机规则如下。

1) 设基础树形中层序每个分支的原编码为code = {0,1,0,1,⋯,0,1}，随机因子为seed。

2) 将code两两分组为$\text{code_pairs}_i = \{0,1\}$，$\text{code_pairs}_i[0]$和$\text{code_pairs}_i[1]$分别对应左右分支的编码。

3) 应用seed后，$\text{code_pairs}_i[0]$在0/1之间随机翻转，概率为50%，若$\text{code_pairs}_i[0] = \{0\}$，则$\text{code_pairs}_i[0] = \{1\}$，若$\text{code_pairs}_i[0] = \{1\}$，则$\text{code_pairs}_i[1] = \{0\}$，直到所有$\text{code_pairs}_i$翻转完毕即路径随机完毕。

编码时参考表2应用随机因子后的路径编码，将待编码信息中的每个字符分别进行编码。采用层序遍历法找出节点中对应字符char_i，记录字符所在节点的路径dist_i，同时生成相应的索引字段index_i，路径dist_i长度不固定但小于或等于4 bit，索引字段长度固定为4 bit。

1) 当字符所在节点深度小于4时，即路径长度小于4，索引字段结构如图6所示。

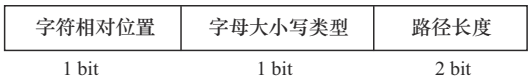


图 6 索引字段结构(深度小于4)

字符相对位置字段代表字符在节点中的位置，可以指示2种不同的位置关系，字母大小写类型字段指示英文字符的大小写类型，路径长度字段用于表示字符所在节点对应的路径长度，在解码时，路径长度是提取各字符对应路径编码的关键。

2) 当字符所在节点深度等于4时，索引字段结构如图7所示。

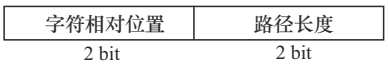


图 7 索引字段结构(深度等于4)

在深度等于4的节点中存放着所有非字母类型字符，此时字母大小写类型字段无效，将其替换为字符相对位置字段，字符相对位置字段可以指示4种不同的位置关系，相应的每个节点中字符数量也增加到4种，可以减少含字符节点的深度，提高

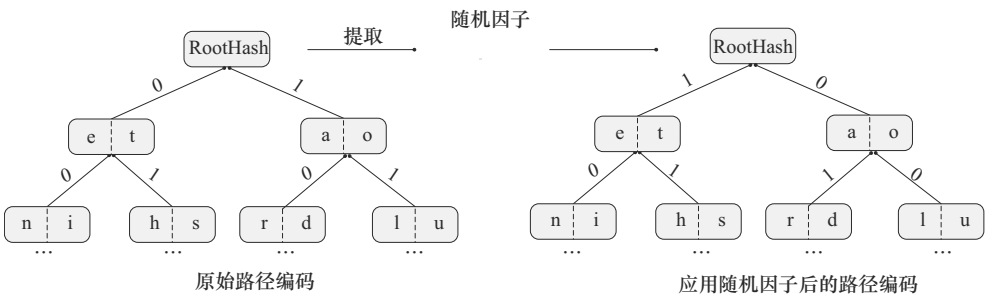


图 5 路径随机前后对比

编码与解码的效率，路径长度字段用于表示字符所在节点对应的路径长度。由于路径长度编码为{1,1}的节点中存放的均为符号类字符，因此规定在编码中不能存在连续2个及以上索引节点中的路径长度编码均为{1,1}的情况。

3.2　信息嵌入

发送方嵌入密文的步骤如下。

1) 为字符 char_i 参照路径编码 dist_i 长度生成 4 bit 的索引字段 index_i，将 dist_i 和 index_i 分别暂存在字符串列表 $\text{dist\_list} = (\text{dist}_0,\text{dist}_1,\cdots,\text{dist}_i)$ 和 $\text{index\_list} = (\text{index}_0,\text{index}_1,\cdots,\text{index}_i)$ 中。

2) 记 N 为所有字符数，$\text{len}(\text{dist\_list})$ 为 l_d，$\text{len}(\text{index\_list})$ 为 l_i。重复执行步骤 1)，直到 $\text{len}(\text{dist\_list}) = \text{len}(\text{index\_list}) = N$。

3) 将 Bin_moment 按位嵌入 dist_i 中，记 $\text{len}(\text{dist}_i) < 4$ 的 char_i 总数为 N_a，若 $\text{len}(\text{dist}_i) < 4$ 则可嵌入 $4 - \text{len}(\text{dist}_i)$ 位，将 Bin_moment 的 17 位完全分布进各 dist_i 中，考虑极限情况：每个字符对应的 $\text{len}(\text{dist}_i) = 1$ 或每个字符对应的 $\text{len}(\text{dist}_i) = 3$ 时，则需要 $\frac{17}{4-1} \leqslant N_\text{a} \leqslant \frac{17}{4-3}$，即 $6 \leqslant N_\text{a} \leqslant 17$。只有当 $N_\text{a} \geqslant 17$ 时，Bin_moment 才可以完全嵌入进各 dist_i 字段中，若 Bin_moment 不能完全嵌入则执行步骤 6)。二进制流时刻嵌入算法伪代码如算法 1 所示。

算法 1　二进制流时刻嵌入算法

输入　Bin_moment，dist_list（未嵌入信息），amount，N_a

输出　dist_list（已嵌入信息）

① if $N_\text{a} \geqslant 17$ then
② 　for $i = 0$ to len(dist_list) − 1 do
③ 　　if 嵌入位 > 17 then
④ 　　break;
⑤ 　　else
⑥ 　　从 Bin_moment 中取出高 $4-\text{len}(\text{dist}_i)$ 位
⑦ 　　将取出的位插入 dist_i 空置位
⑧ 　　end if
⑨ 　end for
⑩ else
⑪ 　将 Bin_moment 插入 amount 字段低 17 位中
⑫ end if

4) 任意生成 2 个路径长度编码均为{1,1}，字符相对位置随机的索引节点编码 $E_\text{a} = \{0/1,0/1,1,1\}$ 和 $E_\text{b} = \{0/1,0/1,1,1\}$，其中 E_a 与 E_b 长度之和共占 1 B。将 dist_list、E_a、E_b 和 index_list 组成编码 secret_code，并记 secret_code 编码长度 $N_\text{e} = \sum_{i=0}^{l_\text{d}} \text{len}(\text{dist\_list}[i]) + \sum_{i=0}^{l_\text{i}} \text{len}(\text{index\_list}[i])$。

由于不同区块链平台数据载荷字段（data payload）最大长度不同，记数据字段长度为 L_data，则根据 N_e 长度分为 2 种情况。

①当 $N_\text{e} \leqslant (L_\text{data} - 1)$ B 时，可以直接进行嵌入。

$$\text{secret\_code} = \sum_{i=0}^{N-1} \text{dist\_list}[i] \| E_\text{a} \| E_\text{b} \| \sum_{i=0}^{N-1} \text{index\_list}[i] \tag{3}$$

②当 $N_\text{e} > (L_\text{data} - 1)$ B 时，将原文均分成 n 片后分批次嵌入，每片中包含的原始信息量几乎相同，并且每片嵌入时均要选择不同的 Bin_moment 执行步骤 3)嵌入每个交易中。

$$\text{secret\_code}_k = \sum_{i=(k-1)\times \text{avg}}^{m_k} \text{dist\_list}[i] \| E_\text{a} \| E_\text{b} \| \sum_{i=(k-1)\times \text{avg}}^{m_k} \text{index\_list}[i] \tag{4}$$

其中，$n = \left\lceil \frac{N_\text{e}}{L_\text{data} - 1} \right\rceil$，$k = 1,2,\cdots,n$，$m_k = \min\{(k \times \text{avg} - 1), N_\text{e} - 1\}$，$\text{avg} = \frac{N_\text{e}}{n}$。

5) 将 $\text{secret\_code} = (b_0,b_1,\cdots,b_{N_\text{e}-1},\cdots,b_{N_\text{e}+7})$ 每 4 位转换为十六进制构成 hex_secret_code，不足 4 位的低位填‘0’，再转十六进制。将 hex_secret_code 嵌入区块链平台的数据载荷字段中，消息嵌入过程结束。

6) 将 Bin_moment 按位嵌入字段长度为 L_amount 的金额字段的低 17 位中，发送交易。

$$\text{amount}(b_{8\times L_\text{amount}-17},b_{8\times L_\text{amount}-16},\cdots,b_{8\times L_\text{amount}-1}) = \text{Bin\_moment}(b_0,b_1,\cdots,b_{16}) \tag{5}$$

3.3　信息提取

如前文所述，消息的发送方和接收方采用同样的规则维护本地的时间型二叉树，并且可恢复出符合 Bin_moment 时刻的树，消息还原步骤如下。

1)分隔路径编码字段和索引节点字段。接收方首先从数据载荷字段中获取密文数据 hex_secret_code′，转二进制 secret_code′，记 secret_code′长度为 N_e'，从后往前逐 4 位遍历找出分隔符 E_a 和 E_b，从而分隔路径编码字段和索引节点字段。

2)提取 Bin_moment′。接收方将索引字段中的各索引节点 index_i' 每 4 位拆分还原为 index_list′。记 index_list′中的 index_i 第一到第四位为 $\{b_{i,0}b_{i,1}b_{i,2}b_{i,3}\}$。遍历 index_list′将每个 index_i 中的 $b_{i,2}b_{i,3}$ 记录在 index_list_temp，取出剩余的路径信息 Bin_dist。

①将所有索引节点低2位转十进制后加1，即 $\text{index\_list\_temp}[i]=\text{BIN2DEC}(\text{index\_list\_temp}[i])+1$。若 index_list_temp 中 $\{b_{i,2},b_{i,3}\}\neq\{1,1\}$ 的数量 $N_a'\geqslant 17$，$\text{index\_list\_temp}[i]=4$，则将索引节点指示的 Bin_dist 中相应的 4 位路径数据放入 dist_list′；若 $\text{index\_list\_temp}[i]<4$，则将 $\text{index\_list\_temp}[i]$ 位数据放入 dist_list′，提取位于当前位置后 4 - $\text{index\_list\_temp}[i]$ 位的数据并按位填充进 Bin_moment′，在 Bin_dist 中 index_i' 对应路径编码中删除同位量的信息，并将删除嵌入信息后的路径放入 dist_list′复执行直到 Bin_moment′长度为 17 位。二进制流时刻提取算法伪代码如算法 2 所示。

算法 2 二进制流时刻提取算法

输入 index_list_temp，Bin_dist

输出 Bin_moment′

① 将 index_list_temp 每个元素转为十进制并加 1
② for $i=0$ to $\text{len}(\text{index\_list\_temp})-1$ do
③ if Bin_moment′长度不足 17 bit then
④ if $\text{index\_list\_temp}[i]==4$ then
⑤ continue;
⑥ else
⑦ 截取从 $\text{Bin\_dist}[i\times 4]$ 到 $\text{Bin\_dist}[(i+1)\times 4]$ 的 4 bit
⑧ 提取这 4 bit 信息中的低 4-$\text{index\_list\_temp}[i]$ 位
⑨ 将提取的低 4-$\text{index\_list\_temp}[i]$ 位填充进 Bin_moment′
⑩ end if
⑪ else
⑫ break
⑬ end if
⑭ end for

②若 index_list_temp 中 $\{b_{i,2},b_{i,3}\}\neq\{1,1\}$ 的数量 $N_a'<17$ 则从交易的金额字段中提取后 17 位的数据得到 Bin_moment′。

3) 由 $\text{Bin\_moment}'(b_0,b_1,b_2,\cdots,b_{16})$ 恢复发送方选择的时刻 selected_moment′，如式(6)所示。

$$\text{selected\_moment}'(\text{hour},\text{minute},\text{second})=\left\{\sum_{i=0}^{4}b_i\times 2^{4-i},\sum_{j=5}^{10}b_j\times 2^{10-j},\sum_{k=11}^{16}b_k\times 2^{16-k}\right\} \tag{6}$$

基于 selected_moment′，使用生成树规则还原出与发送方在同时刻下的相同树形并从根哈希 RootHash′中提取随机因子 seed′，应用 seed′生成与发送方有相同路径编码的时间型二叉树。

4) 采用类似步骤 2)中①的方法进行解码，遍历 index_list 和 index_list_temp 取 $\text{index\_list}[i]$ 和 $\text{index\_list\_temp}[i]$，$i=0,1,2,\cdots,\text{len}(\text{index\_list}')-1$，随后在 Bin_dist′中按顺序依次取出 $(\text{index\_list\_temp}[i][0]\times 2+\text{index\_list\_temp}[i][1]+1)$ 位信息记为 dist_i，按 dist_i 中的位逐位深度搜索二叉树找到字符所在节点，再根据 $\text{index\_list}[i][0]$ 与 $\text{index\_list}[i][1]$ 确定字符相对位置以及类型，此时分为 2 种情况。

①若 $\text{index\_list\_temp}[i]<4$ 或 index_i 中的第三与第四位 $\{b_{i,2},b_{i,3}\}\neq\{1,1\}$，则使用 2.2 节中提出的深度小于 4 的索引字段结构，即 $b_{i,0}$ 表示字符在节点中的相对位置，$b_{i,1}$ 表示字符的大小写，将解码后的字符存入 char_i 中。

②若 $\text{index\_list\_temp}[i]=4$，则使用深度等于 4 的索引字段结构，$b_{i,0}b_{i,1}$ 两位均表示字符在节点的相对位置，将解码后的字符存入 char_i 中。

4 仿真实验

本节通过仿真实验结果分析模型性能。实验平台采用 Windows 系统，CPU 为 I9-13900H@5.40 GHz，GPU 为 Nvidia GeForce RTX 4060，内存大小为 24 GB，网络带宽为 1.2 Gbit/s，选用比特币网络、以太坊和 FISCO BCOS 作为区块链隐蔽通信测试平台。

4.1　发送过程

4.1.1　编码过程

本次实验选取的 message 为 *Jane Eyre* 中的“We had been wandering, indeed, in the leafless shrubbery an hour in the morning”，将其在上述 3 个平台进行模型流程演示。

发送方选取 09:21:17 作为编码时刻 selected_moment 并记录。从选择时刻的时间型二叉树的根节点中获取 RootHash 并从中提取随机因子 seed，则 Bin_moment = (01001010101010001)，具体符号对应的含义和值如表 3 所示。

对 message 的编码参照表 2 中应用随机因子后的路径编码，生成 dist_list，为每个 $dist_i$ 按 2.3 节方法生成对应 $index_i$ 得到 index_list。将 Bin_moment 按 3.2 节步骤 3)嵌入 dist_list 中得到嵌入过时刻信息的 inserted_dist_list。最后执行 3.2 节步骤 4)得到 secret_code（其中包含了 2 个分隔符），将 secret_code 逐 4 位转换为十六进制得到 hex_secret_code 嵌入比特币交易中的 OP_RETURN 字段、以太坊交易 Data 字段或 FISCO BCOS 交易 params 字段后发送。发送过程具体参数如表 4 所示。

4.1.2　上链过程

当 message 编码完毕后需要将结果 secret_code 插入区块链交易的数据载荷字段并发送，在不同区块链平台中插入方法略有不同。

1) 比特币平台

比特币平台的数据载荷字段为 OP_RETURN，其长度为 80 B，采用十六进制格式进行数据存储。

当编码过程结束后，将结果 secret_code 逐 4 位转换为十六进制得到 hex_secret_code，由于长度未

表 3　　**发送过程符号、含义和值**

符号	含义	值
message	密文信息	We had been wandering, indeed, in the leafless shrubbery an hour in the morning
RootHash	根哈希	70ee1e626c88616e6a600589732ddc27afe6a38dfdf4af0392ae90b2e6037eae
seed	随机因子	701626886166600589732276384039290260 37
selected_moment	选择的编码时刻	09:21:17
Bin_moment	二进制编码时刻	01001010101010001
secret_code	二进制编码结果	1100110000011101001011100001111010011…1010 [0111 1011] 0110000000100001000010010010101000000000…0011
hex_secret_code	十六进制编码结果	CC1D261E9…A [7 B] 6021092A00…3

表 4　　**发送过程具体参数**

序号	符号	dist_list	inserted_dist_list	index_list
0	W	110	1100	0110
1	e	1	1100	0000
2		000	0001	0010
3	h	11	1101	0001
4	a	0	0010	0000
5	d	11	1110	1001
6		000	0001	0010
7	b	111	1110	1010
8	e	1	1001	0000
9	e	1	1	0000
⋮			⋮	
len(message) − 1	.	1010	1010	0011

达到80 B，因此可以直接嵌入比特币交易中的OP_RETURN字段，其具体嵌入内容等同hex_secret_code。当信息嵌入完毕后，发送方调用平台接口发送载密交易。

2) 以太坊平台

以太坊平台的数据载荷字段为data，长度受交易费用gas限制，数据采用十六进制格式存储。以太坊平台上链过程与比特币平台相同，将secret_code转换为十六进制表示的hex_secret_code后直接嵌入以太坊交易中的data字段，其具体嵌入内容等同hex_secret_code，最后发送载密交易。

3) FISCO BCOS平台

FISCO BCOS采用智能合约传递信息，嵌入信息长度通常受合约规则和区块大小限制。

本次实验使用较为简化的智能合约进行数据传输，智能合约的发送过程如算法3所示。为了交易成本考虑，本次实验将交易费用设置为零。先对要发送的数据secret_code作为params进行打包得到适合传输的数据格式data，再对data进行签名保证数据的真实性。将发送地址、合约地址、data、gas限制、签名和交易费用全部封装进一个交易对象transaction，最后将交易发送到智能合约地址中。

算法3 智能合约的发送过程

输入 params

输出 txHash

//打包交易数据

1) 打包params得到数据data

//签名打包数据

2) 将data和senderAddress签名

//构造交易对象

3) transaction={

“发送地址”: senderAddress,

//接收地址为合约地址

“接收地址”: contractAddress,

“数据”: data,

“gas限制”: gasLimit,

“签名”: signature,

//交易费用设为零

“交易费用”: 0

}

//发送交易到区块链中的智能合约

4) 发送交易transaction获得交易哈希

5) return交易哈希

4.2 接收过程

4.2.1 读取过程

1) 比特币平台

当交易到达接收方节点后，从交易中的OP_RETURN字段中提取出hex_secret_code′，并转为二进制的secret_code′。

2) 以太坊平台

读取过程同比特币平台，从data字段中提取到hex_secret_code′后转为二进制secret_code′。

3) FISCO BCOS平台

接收方使用智能合约的接收过程如算法4所示，接收方从智能合约中获取交易信息并从中提取数据和签名后，首先验证签名以确保数据的真实性，然后将数据进行解包得到原数据result即secret_code′。

算法4 智能合约的接收过程

输入 transaction

输出 result

//提取交易中相关数据

1) 从交易中获取data

2) 从交易中获取签名signature

3) 验证签名signature

//获取原文

4) 解包data得到原文result

5) return result

4.2.2 解码过程

接收过程符号、含义和值如表5所示。接收方获取二进制的secret_code′后从中找出分隔符得到路径编码字段和索引字段，在索引字段中提取各index_i'组成index_list′，并记录每个index_i'低2位组成index_list_temp。从头遍历index_list_temp找出$\{b_{i,2},b_{i,3}\}\neq\{1,1\}$的元素$\text{index}_i'$，并在Bin_dist中找出索引节点对应的路径编码，从中提取有关Bin_moment′的比特信息并删除，直到提取的信息使Bin_moment′长度达到17 bit，转十进制得到selected_moment′。将所有路径放入inserted_dist_list′，将删除比特信息后的路径dist_i和未嵌入比特信息的路径dist_i均放入dist_list′中。接收方利用selected_moment′还原出发送方选定时刻的时间型二叉树，获取RootHash′并提取出seed′，应用

表5　　**接收过程符号、含义和值**

符号	含义	值
hex_secret_code′	从交易中获取的十六进制密文	CC1D261E9…A [*7 B*] 6021092A00…3
secret_code′	二进制密文	110**01100**0001**1**11**01**00**10**11**10**0001**1**111**0**10011…1010 [0111 1011] 0110000000100001000010010010101000000000…0011
$index_i$′	每个路径编码对应的索引	…
index_list′	从密文中分离出的索引节点列表	[0110,0000,0010,0001,0000,1001,0010,1010,0000,0000,…,0011]
index_list_temp	每个索引低2位的列表	[10,00,10,01,00,01,10,10,00,00,…,11]
Bin_dist	二进制的路径编码字符串（含Bin_moment′嵌入的信息）	110**01100**0001**1**11**01**0**10**11**10**0001**1**111**0**10**01**1…1010
inserted_dist_list′	Bin_dist路径拆分列表	[110**0,**1**100,**000**1,**11**01,**0**010,**11**10,**000**1,**111**0,**1**001,**1,…,1010]
dist_list′	删除 Bin_moment′信息后的路径列表	[110,1,000,11,0,11,000,111,1,1,…,1010]
Bin_moment′	提取的二进制编码时刻	00000000100101010101010001
RootHash′	恢复出的根哈希	70ee1e626c88616e6a600589732ddc27afe6a38dfdf4af0392ae90b2e6037eae
seed′	由根哈希提取出的随机因子	70162688616660058973227638403929026037

seed′后得到与发送方树节点路径完全相同的树。最后执行解码操作，利用 index_list′将 dist_list′解码为原文message。接收过程具体参数如表 6所示。

5　性能分析

本节从比特币平台实验中得到数据，在通信效率、预协商交易取消的影响以及安全性方面对模型的性能进行全面评估。

5.1　效率分析

5.1.1　嵌入率分析

熊礼治等[15]认为基于区块链的隐蔽通信嵌入率为每条交易可以嵌入的秘密信息比特数量，对应本文提出的模型就是发送方每次通信发送的所有交易中嵌入秘密信息的比特总数。

本文采用基于区块链可靠性较强的信息嵌入模型与本文模型进行比较，以保证对比的合理性。

信息嵌入率对比如表7所示，本文模型的信息嵌入率相比于其他模型有较大提升。文献[7]使用动态标签作为信息的载体，取用OP_RETURN字段其中的23字节作为动态标签，嵌入率为184 bit；文献[8]利用马尔可夫链将载密信息转化为自然语言文本，经过大量测试得到最大嵌入率为449 bit；

表6　　**接收过程具体参数**

序号	index_list′	index_list_temp	inserted_dist_list′	dist_list′	符号
0	0110	10	1100	110	W
1	0000	00	1100	1	e
2	0010	10	0001	000	
3	0001	01	1101	11	h
4	0000	00	0010	0	a
5	1001	01	1110	01	d
6	0010	10	0001	000	
7	1010	10	1110	111	b
8	0000	00	1001	1	e
9	0000	00	1	1	e
⋮			⋮		
len(message) − 1	0011	11	1010	1010	.

文献[16]中由于嵌入数据和嵌入消息的α位必须相同，α值的大小会很大程度上影响信息提取的时间，因此α要尽可能小；文献[17]将信息嵌入coinbase中，最大长度69 B，即552 bit；文献[18]中反向DSA通道（RDSAC, reversed-DSA channel）通过私钥生成的伪随机函数（PRF, pseudo random function）嵌入数据，而双DSA通道（DDSAC, double-DSA channel）直接在随机因素中嵌入数据，嵌入率分别为128 bit和256 bit；文献[19]将消息进行分段，每个分段设置为512 bit，因此每笔交易的嵌入率为512 bit；本文模型由于使用到了长度为L_{data} B的不同区块链平台的数据载荷字段并且在一定情况下可能会使用到长度为L_{amount} B的amount字段，因此最大嵌入率为$\left(L_{\text{data}}+L_{\text{amount}}\right)$ B即$8\times\left(L_{\text{data}}+L_{\text{amount}}\right)$ bit。较之前述的方法，本文模型适用于多种具有数据载荷字段的区块链平台而非单一平台，若采用前述方法使用的比特币平台，则本文模型的最大嵌入率为数据载荷字段80 B与金额字段8 B之和，即$8\times(80+8)=704$ bit，嵌入率相较于前述方法分别提升520 bit、255 bit、704−α bit、152 bit、576或448 bit和192 bit。

表7 信息嵌入率对比

实验模型	嵌入率/bit
文献[7]	184
文献[8]	449
文献[16]	α
文献[17]	552
文献[18]	128/ 256
文献[19]	512
本文模型	$8\times\left(L_{\text{data}}+L_{\text{amount}}\right)$=704

5.1.2 传输效率分析

传输时间也是评估模型效果的重要指标之一，指从发送方开始执行对信息的编码操作到接收方正确解码信息并得到原始信息所需要的全部时间。对于本文，传输时间则为编码与发送时间，区块链广播时延以及接收与解码时间的总和。区块链网络的交易发布到确认存在一定的时延，但是，当一个节点生成一笔交易并上传到区块链网络上时，通过洪泛机制在几秒内就会广播全网络[18]。因此，交易确认时间只会较小地影响模型总体传输时间，而信息编码嵌入和密文解码提取的时间长短直接决定了模型时间效率的好坏。

表1中字符出现概率的高低决定了图2时间型二叉树中字符在节点的分布情况。同时本文分别对3本书的文本进行了编码和解码测试，根据统计结果可知本文模型编码和解码时的平均遍历节点数及深度，结果如表8所示。

由于解码采用的是深度搜索方式，因此3本书的文本的平均遍历深度与平均遍历节点数相同，分别为2.10、2.10和2.13。可知在符合自然语言习惯的情况下本文模型平均遍历深度小于3，提高了编码和解码的速度。

下面将从传输小容量信息和大容量信息2种情况对本文模型传输效率进行分析。

1)传输小容量信息

使用本文模型与其他同类基于区块链的隐蔽通信模型进行时间效率的评估，假设固定传输1 KB大小的信息。文献[7]使用载密信息作为随机数，发出交易后，只需几秒时间就可以广播到接收方，而不用等待区块生成，因此传输时间为秒级。文献[8]利用马尔可夫模型进行秘密信息的嵌入与提取，嵌入并提取8 bit的信息量需要平均花费3 s的

表8 编码与解码测试

测试类型	指标	*Jane Eyre*	*Pride And prejudice*	*Oliver Twist*
编码	平均遍历节点数	7.52	7.23	7.44
	平均遍历深度	2.10	2.10	2.13
	时间/s	1.21	0.81	1.02
解码	平均遍历节点数	2.10	2.10	2.13
	平均遍历深度	2.10	2.10	2.13
	时间/s	0.88	0.61	0.78

时间，因此传输总时间超过 10 min。文献[9]采用图像隐写和链下 IPFS 存储相结合的方式，在提高嵌入率的同时，降低了传输速度，传输时间为分钟级。文献[16]需要生成区块后才能提取嵌入的信息，每个区块至多传输 $K\alpha$ bit，K 为交易数，因此传输远大于 10 min。文献[18]有 RDSAC 和 DDSAC 这 2 种方法，RDSAC 利用私钥加密信息并使用随机因素作为标签并放入交易，接收者过滤特殊交易并使用伪随机函数和椭圆曲线来识别含密交易，由于需要计算椭圆曲线，总时长大于 1 min；DDSAC 则将隐秘数据直接嵌入交易中的随机因素，并从正常交易中抽样交易金额作为标签，避免了 RDSAC 中的传输时延，且未采用椭圆曲线识别交易，总时长小于 1 min。文献[19]需要扫描区块中的交易并对比标记，并且传输时间随着消息分段的增加而变长，因此传输时间在测试网和主网内均为分钟级。

本文编码和解码 1 KB 大小的信息时间均为毫秒级，加入必要的区块链广播时间后传输时间变为秒级。本文模型与文献[7]和文献[18]提出的隐蔽通信模型的传输时间虽然均为秒级，但编码与解码效率都较之更高，其他模型的传输时间均为分钟级，因此本文模型较之前述所有模型传输效率都更高。传输小文本时间对比如表 9 所示。

表 9　　传输小文本时间对比

实验模型	传输时间量级
文献[7]	秒级
文献[8]	分钟级
文献[9]	分钟级
文献[16]	分钟级
文献[18]	分钟级/秒级
文献[19]	分钟级
本文模型	秒级

2) 传输大容量信息

假设采用随机生成的不同大小的文本作为传输测试文本（共 20 段，每个文本间隔大小 50 KB），得到本文模型编码时间和解码时间随文件大小变化的总体趋势，如图 8 所示。

由图 8 可知，信息容量小于 1 000 KB 的文本的编码与解码时间均小于 1 s，且编码与解码时间同文本大小的变化呈线性相关。

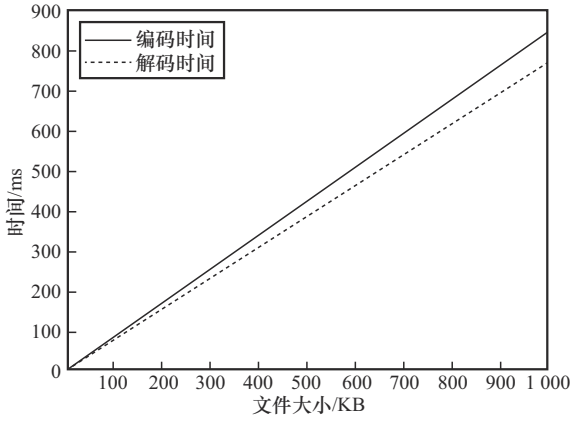


图 8　编码和解码时间

在传输更大容量信息时，为考虑成本问题，采用 FISCO BCOS 平台进行传输，在该平台中通过智能合约传递数据，合约中设置数据载荷字段容量小于区块大小（4 MB）。以上述统计的编码与解码时间结果为基础，现对大小为 100 MB 原信息分别以 1 MB、2 MB 和 4 MB 为单位分片后发送，统计信息从开始编码到解码完毕的总时间（交易广播时间取平均值 7 s）。将本文模型的统计结果与适用于大容量信息传输的模型对比，结果如表 10 所示。

表 10　　传输大文本时间对比

模型	分片长度	传输时间/s
文献[9]	无	>3 164
文献[19]	64 B	>3 258
本文模型	1 MB	453
本文模型	2 MB	440
本文模型	4 MB	434

文献[9]使用链下存储方式传输大容量信息，使用链下 IPFS 存储可以一次性存储较大容量的数据，因此不需要对信息分片，但该模型的传输时间会受网络环境和通信环节增加导致的传输时延影响，并且使用了图像隐写技术进行信息嵌入，因此传输总时间较长。文献[19]采用动态标签生成技术和对称加密技术导致计算开销较大，且规定单笔交易至多嵌入 512 bit 即 64 B，导致分片数较多，因此传输时间较长。本文采用时间型二叉树对信息进行编码与解码，从数据结构层面提高了编码与解码效率，并且不额外增加通信环节，缩短了整体传输时间。因此本文模型较之前所述适用于大容量信息传输的模型传输效率更高。

5.2 安全性分析

5.2.1 抗频率分析

频率分析是通过统计字母或符号出现的频率来破解加密信息的一种方法[20]，通常用于固定长度编码。对于本文方案，字符的编码长度在1~4位并且密文中包含索引节点字段，此外，编码之间的分布是经过频率平衡的，上述这些特性增加了统计分析的复杂度。

攻击者需要统计路径编码字段中的编码分布频率反推信息，理论上若攻击者知道采用树形编码的方式，可以根据所有编码的频率分布，建立一个编码频率二叉树反推出可能对应的字符。但攻击者获取的密文中有索引节点字段的存在，破坏了原有路径编码的频率分布，使攻击者无法正常分析。本文设计了一个采用频率分析的攻击实验，假设攻击者获知编码是采用树形结构，但不知道字符在树中的排列顺序且无法排除索引节点字段，原文message为“We had been wandering, indeed, in the leafless shrubbery an hour in the morning”。

攻击者获知编码采用树形结构，则可知字符对应编码长度为1~4位。首先从区块链中获取编码信息转为二进制密文secret_code′，使用大小为1~4位的窗口遍历secret_code′统计所有编码情况出现的频率，过程如图9所示，通过编码频率分布构建可能的频率二叉树，反推出可能的字符。

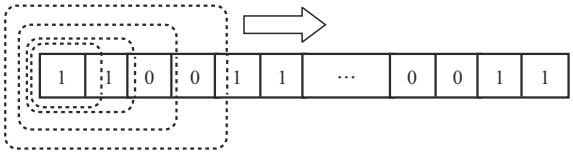


图9 编码频率统计过程

将统计结果与message真实编码占比进行对比，结果如图10所示。使用K-S检验（Kolmogorov-Smirnov检验）可以判断2个样本的概率分布是否不同[21]，K-S统计量的数学定义为$D=\max|F_1(x)-F_2(x)|$，$F_1(x)$和$F_2(x)$分别是统计编码分布和实际编码分布累积分布函数，使用0.05作为显著性水平来判断检验结果的显著性。经计算得到$D=0.45$，$p=0.002\,5<0.05$，即统计编码分布和实际编码分布来自不同的分布，攻击方无法通过频率统计的方式破译编码反推出message。

此外，将同深度字符采用平衡频率分布，不仅可以减少同深度某个节点路径出现频率明显高于同深度其他节点的情况，还增加了统计分析的复杂度。记图2中根节点的左子节点号为1，层序法依次编号得到所有包含字母类字符的节点编号依次为1~13，统计文本*Jane Eyre*编码时各节点路径出现频率如图11所示。

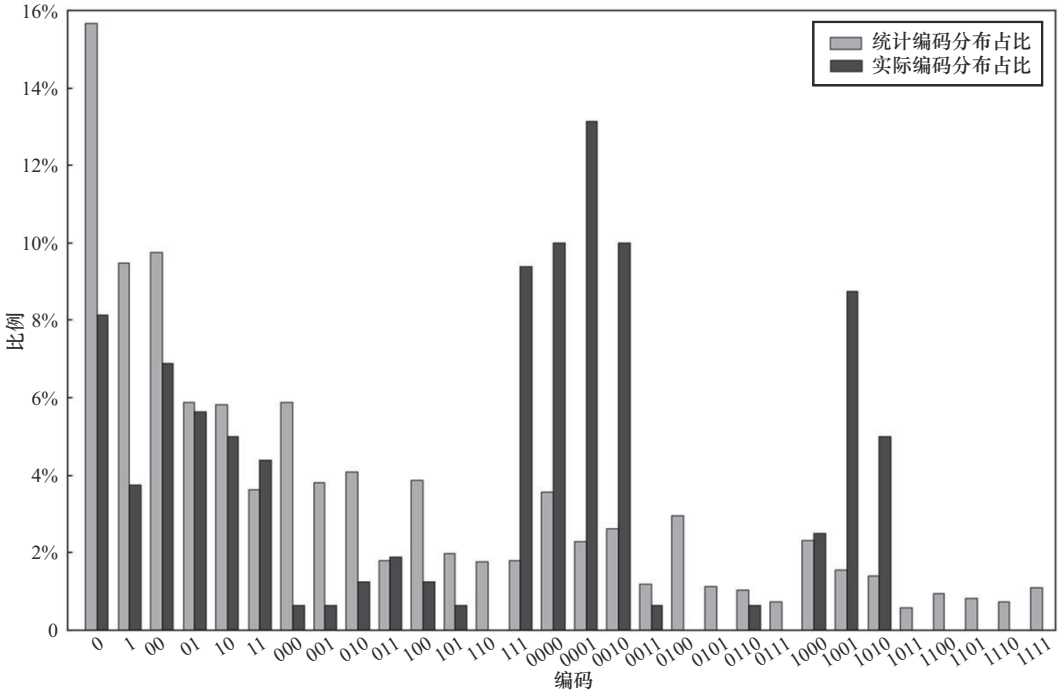


图10 统计编码与实际编码比例对比

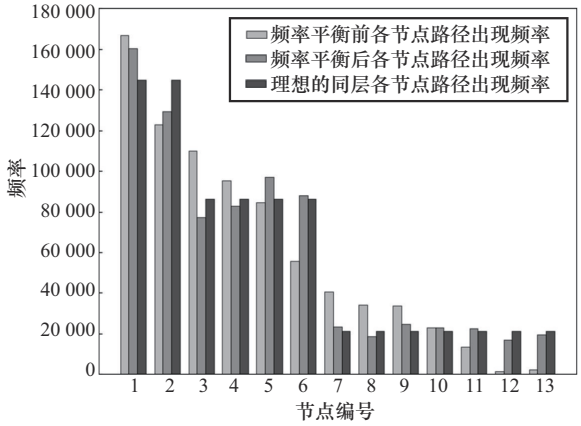


图 11　节点路径出现频率

信息熵为不确定性的度量，信息熵越大则不确定性越高[22]，定义为 $H(X)=-\sum_{i=1}^{n}p(x_i)\mathrm{lb}\left(p(x_i)\right)$，$p(x_i)$表示随机事件$X$为$x_i$的概率。表 11 为图 11 中 3 种情况对应的信息熵。

表 11　　信息熵比较

频率平衡情况	信息熵
频率平衡前	3.185 0
频率平衡后	3.283 8
理想频率	3.291 2

理想频率是指利用本文频率统计的方法确定字符所在树的深度，相同深度的字符通过调换分布的节点，达到同深度的节点编码出现频率相同的状态。例如节点 1 与节点 2 在同一深度，理想的编码频率是相同的。

可以看出，频率平衡后的信息熵接近理想频率的信息熵，并且大于平衡前的信息熵，即频率平衡后信息的不确定性增大，并趋近于本文模型的理想状态，有助于增强抗频率分析的性能。

5.2.2　*抗暴力破解性能*

时间型二叉树的前 4 层的字符排列共有 $A_{92}^{92}=92!$ 种可能，通过暴力破解字符排列顺序几乎不可能。并且随机因子是从根哈希中提取的，随着时间的改变树会不断生成新的叶子节点，根哈希会随时间的变化而改变，即随机因子也是随时间不断变化的，又通过随机因子维护相同的字符−路径映射关系，因此字符−路径映射关系会随时间改变。随着节点数量的增加，根哈希的破译难度呈指数级上升，使用 SHA-256 进行每次的根哈希运算时，每增加一个节点需要多 2^{128} 次运算。当攻击者破译时间大于节点生成时间时，映射关系就具有不可破解性。

5.2.3　*多组并发通信分析*

在大规模通信场景时，不可避免地会有 2 种情况出现。

1) 多条来自不同发送方的载密交易在较短的时段内到达同一接收方地址的情况。

2) 发送方同时向不同接收方发送信息的情况。

针对情况 1)，模型首先接收多平台的信息并对来自同地址的交易进行归类，再对同地址的多笔交易进行 selected_moment 提取并排序，参照 5.2.4 节的方法判断是消息分片还是其他消息再进行分类，最后对所有分类后的消息执行解码操作得到密文。本文针对情况 1)设计并模拟了如下实验，假设地址 R 为接收方地址记为 $\mathrm{address}_R$，在较短时间段内同时接收了多条来自不同发送地址 $\mathrm{address}_{S_i}$ 的载密交易，其中包含同地址发送的消息分片交易和其他消息交易。对比正常消息按序到达并解码以及情况 1) 的 2 种状态下的模型解码准确率，由于存在一定概率受网络影响导致交易无法正常发送或接收，设该概率为 p_e，结果如表 12 所示。

表 12　　解码准确率对比

通信状态	比特币平台	FISCO BCOS 平台
正常	$100\%-p_e$	$100\%-p_e$
多组并发(<12.5 s)	$87.5\%-p_e$	$93\%-p_e$
多组并发(≥12.5 s)	$100\%-p_e$	$100\%-p_e$

忽略网络影响时，在双平台中正常按序到达的消息均可以正确解码。而在多组并发通信时，若接收时间窗口小于 12.5 s，非分片消息可以正确解码，而某些载有分片消息的交易未能及时到达导致消息不完整，无法正常解码。FISCO BCOS 平台单笔交易信息载荷多，消息分片数量少，并且交易广播时间较比特币平台短，因此 FISCO BCOS 平台解码准确率较比特币平台高。当接收时间窗口大于 12.5 s 时，几乎所有发送的消息可以广播至全网络[15]，分片消息可以进行完整解码。

针对情况 2)，当同时向不同接收方发送信息时，发送方首先为所有待发信息选择发送平台，若信息长度超过比特币平台载荷字段长度则选择 FISCO BCOS 平台进行发送降低通信成本，若长度

仍超出FISCO BCOS平台的载荷字段长度，则将信息分片，并参照5.2.4节的方法确保消息连续性。当所有待发信息的发送平台选择完毕且分片完毕后，采用多线程并发的方式向不同接收方发送相应信息。情况2)的平台选择方式和分片规则均与正常通信一致，因此在准确率方面与正常通信无明显差别。

5.2.4 消息连续性分析

当消息要分片发送时，区块链广播的时延会导致多条交易信息到达接收方的时间不固定，造成消息接收顺序混乱。发送方在每次传递分片消息message_i时均会选择连续的或间隔极短的时刻，即$0 < \text{selected_moment}_i - \text{selected_moment}_{i-1} \leqslant T_s$，例如$0 < T_s \leqslant 5$ s，并且在每片信息编码中分别嵌入这些时刻，使接收方利用时序恢复正确顺序的密文信息。

而选择前后间隔较长的时刻$\text{selected_moment}_i - \text{selected_moment}_{i-1} \geqslant T_l$，例如$T_l \geqslant 1$ h，则可以区分消息的时效性，确保新消息和旧消息之间有明确的时间分隔。

5.3 预协商简化分析

本节将从隐蔽性和通信效率分析取消预协商交易对通信的安全性和效率进一步提升的效果。

5.3.1 隐蔽性分析

预协商交易通常会包含密钥、分片等关键信息，预协商交易的发送会影响交易间隔的分布并且会引起节点流量的改变，因此该机制增加了隐蔽通信被探测概率。

本文收集了比特币网络内近期约1 500万条交易的交易时间，过滤只出现过一次的交易地址和交易过于频繁的大商交易地址，得到1 h内同地址再次交易的间隔统计。

假设同地址在1 h内发送了n笔交易$\{T_1,T_2,\cdots,T_n\}$，则同地址发送交易间隔为计算相邻2个交易的时间戳（Unix为时间戳的毫秒数格式）差值除以60 000后向上取整，如式(7)所示。

$$\text{Transaction\_Interval}=\left\lceil \frac{\text{Time}_{T_2}-\text{Time}_{T_1}}{60\,000},\frac{\text{Time}_{T_3}-\text{Time}_{T_2}}{60\,000},\cdots,\frac{\text{Time}_{T_n}-\text{Time}_{T_{n-1}}}{60\,000} \right\rceil \tag{7}$$

图12为1 h内同地址发送交易次数小于或等于5次的统计。

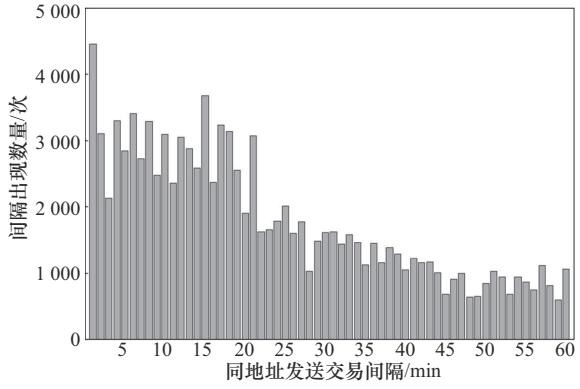


图12 正常同地址交易间隔

为了印证预协商交易会增加隐蔽通信被探测概率，本文大致遵循图12的发送交易间隔发送了4 000条交易，其中1 000条是采用本文模型取消预协商的交易，另外3 000条交易是在每条正式通信交易之前包含额外发送的一次预协商交易。

图13(a)为取消预协商交易直接发送隐蔽交易的情况，共1 000条均遵循发送交易间隔规律；图13(b)~图13(d)为采取不同发送策略的隐蔽交易间隔分布情况，但都遵循发送交易间隔规律。图13(b)为预协商交易后短时间内再次发送隐蔽交易的策略，保证通信效率；图13(c)为预协商后在合理的时间（20 min）内再次发送隐蔽交易的策略，兼顾通信效率和隐蔽性的通信策略；图13(d)为预协商后完全按发送间隔规律再次发送隐蔽交易的策略，该策略完全考虑隐蔽性。

K-L散度（Kullback-Leibler散度）是2个概率分布间差异的非对称性度量，定义为

$$D_{\text{KL}}(P\|Q)=\sum_x P(x)\text{lb}\left(\frac{P(x)}{Q(x)}\right) \tag{8}$$

其中，P表示数据的真实分布，Q表示数据的理论分布。使用K-L散度可以度量取消预协商交易前后的发送交易间隔分布差异度，根据分布差异度的大小可以判断交易隐蔽性的强弱[23]。

记图12的分布为Q，图13(a)~图13(d)的分布分别为P_a、P_b、P_c、P_d，则通过定义可得对应的K-L散度如式(9)~式(12)所示，计算结果如表13所示。

$$D_{\text{KL}_a}(P_a\|Q)=\sum_{x=1}^{60} P_a(x)\text{lb}\left(\frac{P_a(x)}{Q(x)}\right) \tag{9}$$

$$D_{\text{KL}_b}(P_b\|Q)=\sum_{x=1}^{60} P_b(x)\text{lb}\left(\frac{P_b(x)}{Q(x)}\right) \tag{10}$$

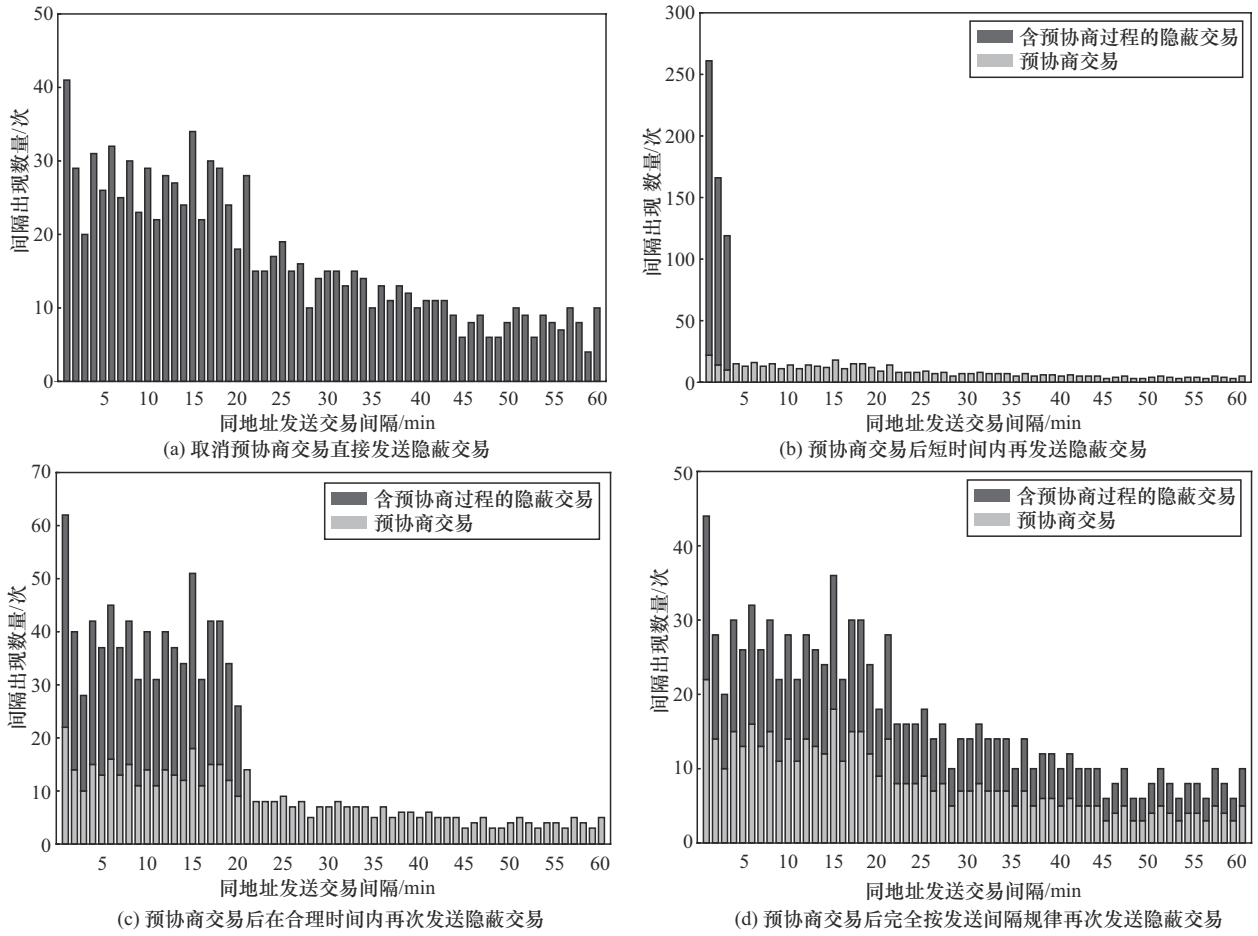


图 13　预协商交易取消前后发送交易间隔分布

$$D_{\mathrm{KL_c}}\left(P_{\mathrm{c}}\|Q\right)=\sum_{x=1}^{60}P_{\mathrm{c}}(x)\mathrm{lb}\left(\frac{P_{\mathrm{c}}(x)}{Q(x)}\right)\tag{11}$$

$$D_{\mathrm{KL_d}}\left(P_{\mathrm{d}}\|Q\right)=\sum_{x=1}^{60}P_{\mathrm{d}}(x)\mathrm{lb}\left(\frac{P_{\mathrm{d}}(x)}{Q(x)}\right)\tag{12}$$

表 13　不同发送策略对应的 K-L 散度

发送策略的分布	K-L 散度
P_{a}	0.000 41
P_{b}	0.669 87
P_{c}	0.114 34
P_{d}	0.000 87

由表 13 可以看出，除了取消预协商交易的交易分布与原始发送交易间隔分布差异较小外，有预协商交易的发送交易间隔分布 P_{b} 和 P_{c} 均与原始分布有较大差异，隐蔽性无法保证，P_{d} 和 P_{a} 虽然分布类似且都与原始分布差异较小，但 P_{d} 传输的信息量只有 P_{a} 的 50%，因此取消每次发送隐蔽交易前的预协商交易可以提高通信过程的隐蔽性即提高安全性。

5.3.2　通信效率分析

预协商交易的发送不仅会减少通信的隐蔽性还会增加传输真实数据量的总时间，表 14 为采用上述 4 种发送策略发送相同信息量所需时间。

表 14　发送相同信息量所需时间

发送策略的分布	时间/s
P_{a}	22.34
P_{b}	24.04
P_{c}	32.41
P_{d}	44.61

经测算，若要传输与 P_{a} 发送方式相同的信息量，P_{b} 和 P_{c} 发送方式分别要多花费 7% 和 45% 的时间；P_{d} 和 P_{a} 发送方式虽然分布类似且都与原始分布差异较小，但 P_{d} 发送方式只有一半的交易是用于隐蔽通信，即要多花费近 100% 的通信时间才能传输与 P_{a} 发

送方式相同的信息量。因此取消每次发送隐蔽交易前的预协商交易可以缩短发送时间，提高通信效率。

5.4 通信成本分析

实验测试的硬件平台同第 4 节仿真实验平台。

1) 算力耗费

对多段总大小为 100 MB 的文本进行编码和解码时的算力监控，得到 CPU 平均运行频率、内存使用峰值以及 GPU 使用率的统计结果，如表 15 所示。

表 15 算力耗费统计

过程	CPU 平均运行频率/MHz	内存使用峰值/MB	GPU 使用率
编码过程	471	1 851	1%
解码过程	553	15 491	2%

由表 15 可以看出，解码过程在内存使用中远高于编码过程，这是由于在解码时需要将大量的二进制类型的密文存入内存中，而编码过程则是将待编码字符存入内存中；2 个过程 CPU 平均运行频率没有明显差别，分别占 CPU 总频率的 8.5% 和 10%；GPU 使用率分别为 1% 和 2%，说明模型几乎不会使用到 GPU 进行计算。因此，本文模型对算力的需求较低，可以节省算力成本。

2) 网络带宽占用分析

动态二叉树的生成和维护在各通信方的本地进行，无须进行网络交互，但交易的发送与接收需要占用一定的网络带宽。

在比特币平台中，每次发送的内容包含比特币的 P2P 协议头（长度 24 B）和交易数据（长度 274 B，其中 OP_RETURN 占 80 B），因此发送单笔交易约消耗 298 B 的网络流量；在以太坊平台中，每次发送内容包含以太坊的 P2P 协议（长度 50 B）和数据字段（包含长度为 184 B 的标准交易和长度为 m B 的 data 字段），因此发送单笔交易大约消耗（234+m）B 的网络流量；在 FISCO BCOS 平台中，协议头长度为 100～300 B，而 data 字段长度是由智能合约设置的，最大长度小于区块大小 4 MB，因此发送一个交易消耗的网络流量至多为一个区块大小 4 MB，具体结果如表 16 所示。

本文实验平台网络带宽为 1.2 Gbit/s，符合日常使用标准，由表 16 可知，在比特币和以太坊平台中，发送单笔载密交易占用的网络带宽远小于 1%；对于 FISCO BCOS 平台，假设单笔交易长度取最大值 4 MB，则发送单笔载密交易仅占网络带宽的 2.6%。因此，即使多笔数据长度较长的交易同时收发也不会占用过多的网络带宽。由于交易接收的数据包与交易发送的数据包几乎相同则交易接收占用的网络带宽与交易发送的几乎相同。

表 16 发送单笔载密交易流量耗费分析

平台	流量消耗情况
比特币	298 B
以太坊	(234+m) B
FISCO BCOS	<4 MB

因此在进行大规模通信时，本文模型可以将交易收发对网络带宽的占用维持在相对较低的水平，从而产生较低的网络通信成本。

6 结束语

本文针对现有区块链隐蔽通信存在的问题，提出了一种基于动态时间型二叉树的隐蔽通信模型。该模型的特点主要有：使用树形数据结构提高了编码与解码效率，并使用时间作为关键信息确保通信方之间可以还原出相同的树形，在保证一定的安全性的同时提升了嵌入率和编码解码效率；利用根哈希作为随机因子载体，确保了树节点路径编码的一致；将时间嵌入路径编码的空置位中传输，消除了每次通信前的预协商交易，通信的效率和交易的隐蔽性得到进一步提升。实验结果验证了所提模型的有效性。

下一步的研究方向是将该模型用于多组并发的多对多通信之中，并研究抗干扰策略，进一步提高通信效率和安全性。

参考文献：

[1] BERNSTEIN D J, BREITNER J, GENKIN D, et al. Sliding right into disaster: left-to-right sliding windows leak[C]//Proceedings of International Conference on Cryptographic Hardware and Embedded Systems. Berlin: Springer, 2017: 555-576.

[2] ZHANG T, WU Q H, WANG Q, et al. Covert communication via blockchain: hiding patterns and communication patterns[J]. Computer Standards & Interfaces, 2024, 90: 103851.

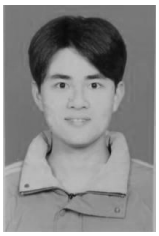
[3] 黄冬艳, 李琨. 多地址的时间型区块链隐蔽通信方法研究[J]. 通信学报, 2023, 44(2): 148-159.
HUANG D Y, LI K. Research on multi-address time-based blockchain covert communication method[J]. Journal on Communications, 2023, 44(2): 148-159.

[4] ZHANG T, LI B Y, ZHU Y, et al. Covert channels in blockchain and blockchain based covert communication: overview, state-of-the-art, and future directions[J]. Computer Communications, 2023, 205: 136-146.

[5] ZHANG J, ZHONG S, WANG T, et al. Blockchain-based systems and applications: a survey[J]. Journal of Internet Technology, 2020, 21(1): 1-14.
[6] PARTALA J. Provably secure covert communication on blockchain[J]. Cryptography, 2018, 2(3): 18.
[7] TIAN J, GOU G P, LIU C, et al. DLchain: a covert channel over blockchain based on dynamic labels[C]//Information and Communications Security. Berlin: Springer, 2020: 814-830.
[8] 佘维, 荣欣鹏, 刘炜, 等. 基于马尔可夫链的生成式区块链隐蔽通信模型[J]. 通信学报, 2022, 43(10): 121-132.
SHE W, RONG X P, LIU W, et al. Generative blockchain-based covert communication model based on Markov chain[J]. Journal on Communications, 2022, 43(10): 121-132.
[9] 佘维, 霍丽娟, 刘炜, 等. 一种可隐藏敏感文档和发送者身份的区块链隐蔽通信模型[J]. 电子学报, 2022, 50(4): 1002-1013.
SHE W, HUO L J, LIU W, et al. A blockchain-based covert communication model for hiding sensitive documents and sender identity[J]. Acta Electronica Sinica, 2022, 50(4): 1002-1013.
[10] NAKAMOTO S. Bitcoin: a peer-to-peer electronic cash system[R]. 2008.
[11] 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.
[12] XIAO Y, ZHANG N, LOU W J, et al. A survey of distributed consensus protocols for blockchain networks[J]. IEEE Communications Surveys & Tutorials, 2020, 22(2): 1432-1465.
[13] MERKLE R C. A digital signature based on a conventional encryption function[C]//Conference on the Theory and Application of Cryptographic Techniques. Berlin: Springer, 1988: 369-378.
[14] 何蒲, 于戈, 张岩峰, 等. 区块链技术与应用前瞻综述[J]. 计算机科学, 2017, 44(4): 1-7, 15.
HE P, YU G, ZHANG Y F, et al. Survey on blockchain technology and its application prospect[J]. Computer Science, 2017, 44(4): 1-7, 15.
[15] 熊礼治, 朱蓉, 付章杰. 基于交易构造和转发机制的区块链网络隐蔽通信方法[J]. 通信学报, 2022, 43(8): 176-187.
XIONG L Z, ZHU R, FU Z J. Covert communication method of blockchain network based on transaction construction and forwarding mechanism[J]. Journal on Communications, 2022, 43(8): 176-187.
[16] SONG S, PENG W. BLOCCE +: an improved blockchain-based covert communication approach[J]. Journal of Chongqing University of Technology (Natural Science), 2020, 34(9): 238-244.
[17] 吕婧淑, 操晓春. 基于比特币系统的隐蔽通信技术[J]. 信息安全学报, 2021, 6(2): 143-152.
LYU J S, CAO X C. Covert communication technology based on Bitcoin[J]. Journal of Cyber Security, 2021, 6(2): 143-152.
[18] CHEN Z, ZHU L H, JIANG P, et al. Exploring unobservable blockchain-based covert channel for censorship-resistant systems[J]. IEEE Transactions on Information Forensics and Security, 2024, 19: 3380-3394.
[19] ZHANG C, ZHU L H, XU C, et al. EBDL: Effective blockchain-based covert storage channel with dynamic labels[J]. Journal of Network and Computer Applications, 2023, 210: 103541.
[20] DUNMORE A, SAMANDARI J, JANG-JACCARD J. Matrix encryption walks for lightweight cryptography[J]. Cryptography, 2023, 7(3): 41.
[21] ZHU L H, LIU Q, CHEN Z, et al. A novel covert timing channel based on Bitcoin messages[J]. IEEE Transactions on Computers, 2023, 72(10): 2913-2924.
[22] LIU F, FAN H Y, QI J Y. Blockchain technology, cryptocurrency: entropy-based perspective[J]. Entropy, 2022, 24(4): 557.
[23] 李雷孝, 杜金泽, 林浩, 等. 区块链网络隐蔽信道研究进展[J]. 通信学报, 2022, 43(9): 209-223.
LI L X, DU J Z, LIN H, et al. Research progress of blockchain network covert channel[J]. Journal on Communications, 2022, 43(9): 209-223.

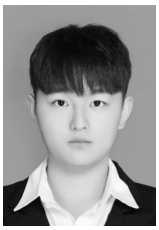
[作者简介]

佘维（1977– ），男，湖南常德人，博士，郑州大学教授、博士生导师，主要研究方向为区块链技术、信息安全、智能系统。



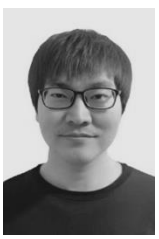
马佳伟（2000– ），男，河南商丘人，郑州大学硕士生，主要研究方向为区块链技术、信息安全。

张淑慧（1999– ），女，湖北襄阳人，郑州大学硕士生，主要研究方向为区块链技术、信息安全。



程孔（2000– ），男，湖南长沙人，郑州大学硕士生，主要研究方向为区块链技术、信息安全。

刘炜（1981– ），男，河南安阳人，博士，郑州大学副教授、博士生导师，主要研究方向为区块链技术、信息安全、智慧医疗。



田钊（1985– ），男，河南荥阳人，博士，郑州大学副教授、硕士生导师，主要研究方向为区块链技术、信息安全、智能交通。