高级脑实战演示教程

达哥（@btcdage）丨2025 年 08 月 19 日

nostr：npub17ahz4xa3hvkvvhh4wguzzqknp8p7l5nyzzqc3z53uq538r5qgn0q40z7pw

这是一个关于“高级脑钱包”的实战演示教程。所谓“高级脑钱包”，其核心思想在于通过一套自己独创的、可重复的算法，将一个容易记忆的“脑口令”和一个绝对私密的“盐值”结合，生成确定性的钱包地址。这种方式兼顾了记忆的便捷性和资产的极致安全性。

步骤如下：

一、 设计高级脑算法

这个教程只演示一种算法思路，你可以基于此原则，创造出千变万化、只属于你自己的算法。

1、脑口令 A

使用任何一个支持 BIP39 算法的钱包（如 Electrum、imToken 等）生成一套标准的 12 或 24 个单词的助记词。这套助记词将作为我们算法的“脑口令”，它可以进行物理备份，甚至可以公开。

2、盐值 B 和哈希次数 N

选用一个只有你自己能记住、对他人毫无意义的字符串（建议长度≥20）作为“盐值”。同时，设定一个哈希迭代的次数。这两个参数是绝对的隐私，绝不记录在任何物理或电子介质上。使用我们的脑钱包工具对“脑口令”和“盐值”进行指定次数的哈希运算，最终生成钱包地址。

二、 选择工具

1、 生成脑钱包工具：

我们使用达哥开源的比特币脑钱包生成器 V0.2.1。

开源地址： https://github.com/btcdage2000/BrainWalletGenerator/

可执行文件下载地址：

https://github.com/btcdage2000/BrainWalletGenerator/releases/tag/0.2.1

此工具的核心功能是：将“脑口令”和“盐值”作为输入，进行多次 SHA256 哈希运算，生成最终的种子，并由此派生出私钥和地址。

2、 轻钱包：

我们使用 Electrum 的一个 fork 版本 NextBTC Wallet 进行演示。

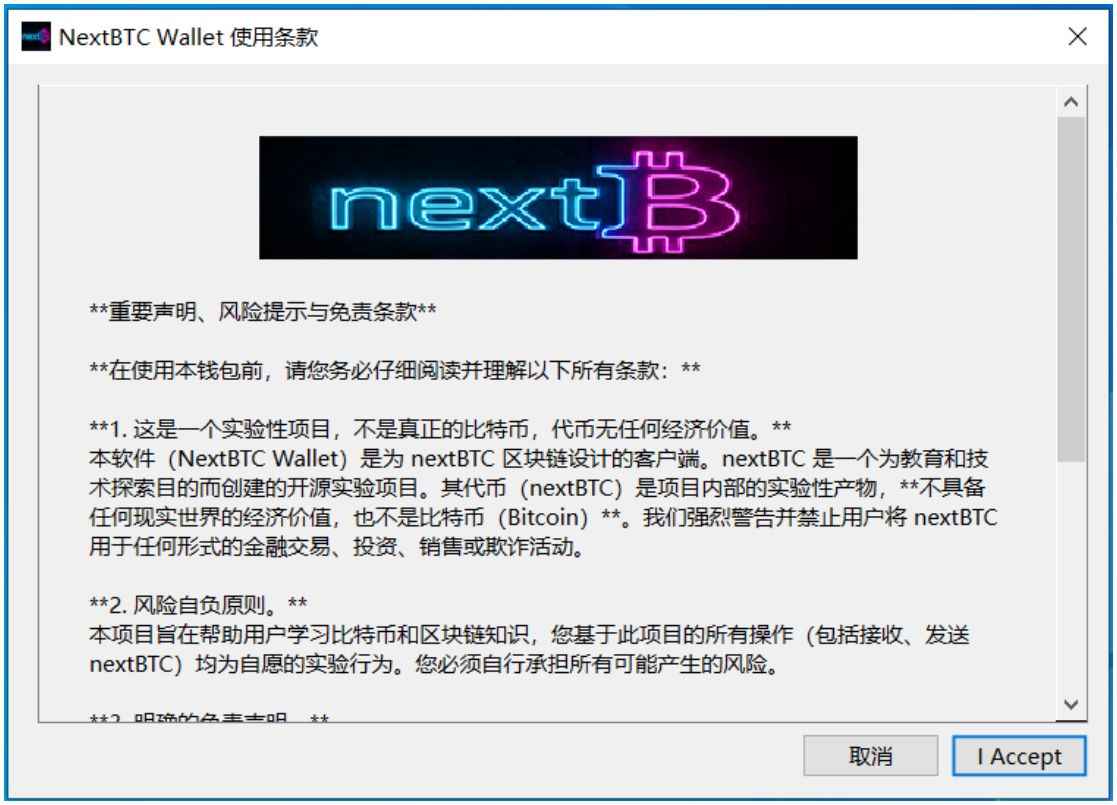
它的功能主要有两个：

生成一套 BIP39 助记词，作为我们算法的初始“脑口令 A”。

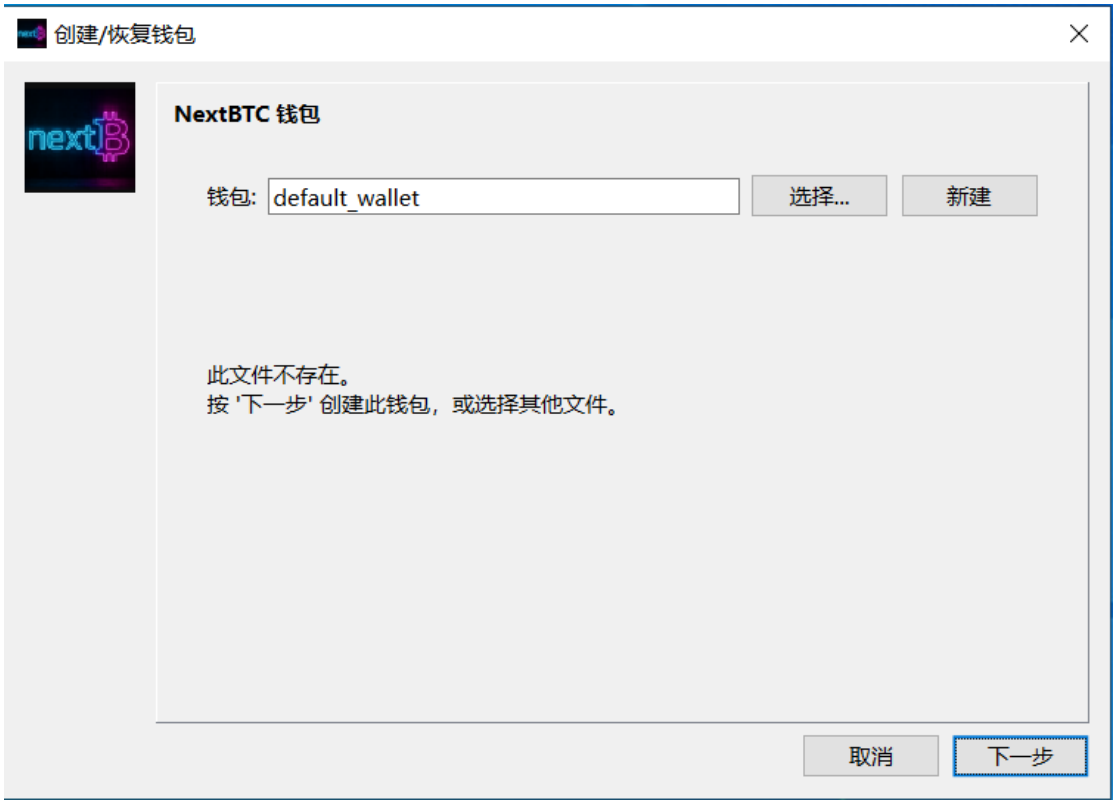
在需要转账时，导入最终生成的私钥，进行地址验证和交易签名。

三、 实战过程：

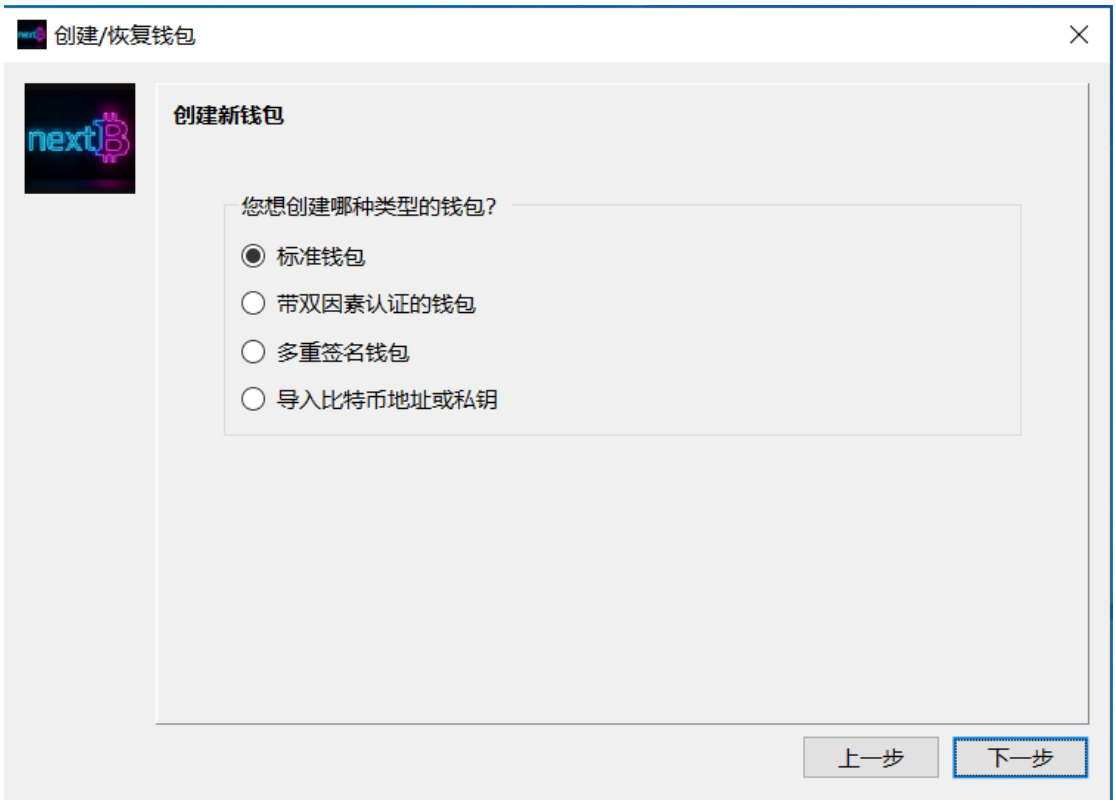
运行 Electrum 钱包（我们演示时使用 NextBTC Wallet），利用它来产生一套随机的助记词。



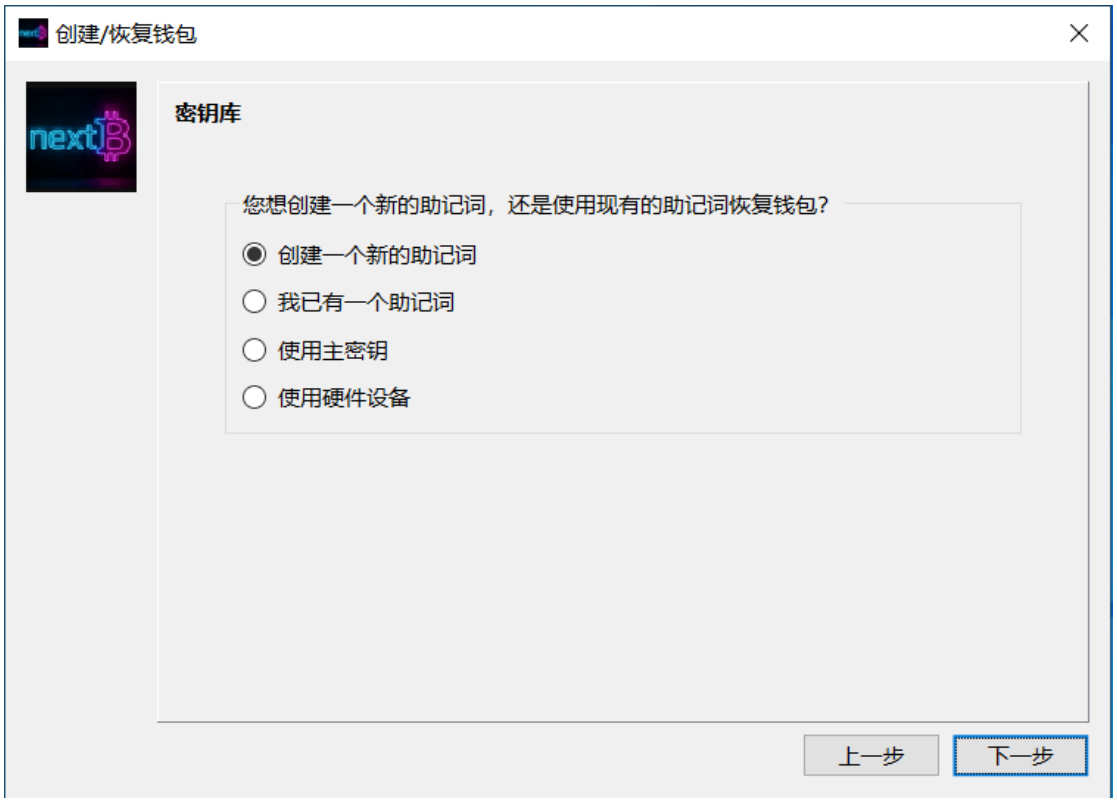
接受使用条款，进入钱包创建界面。

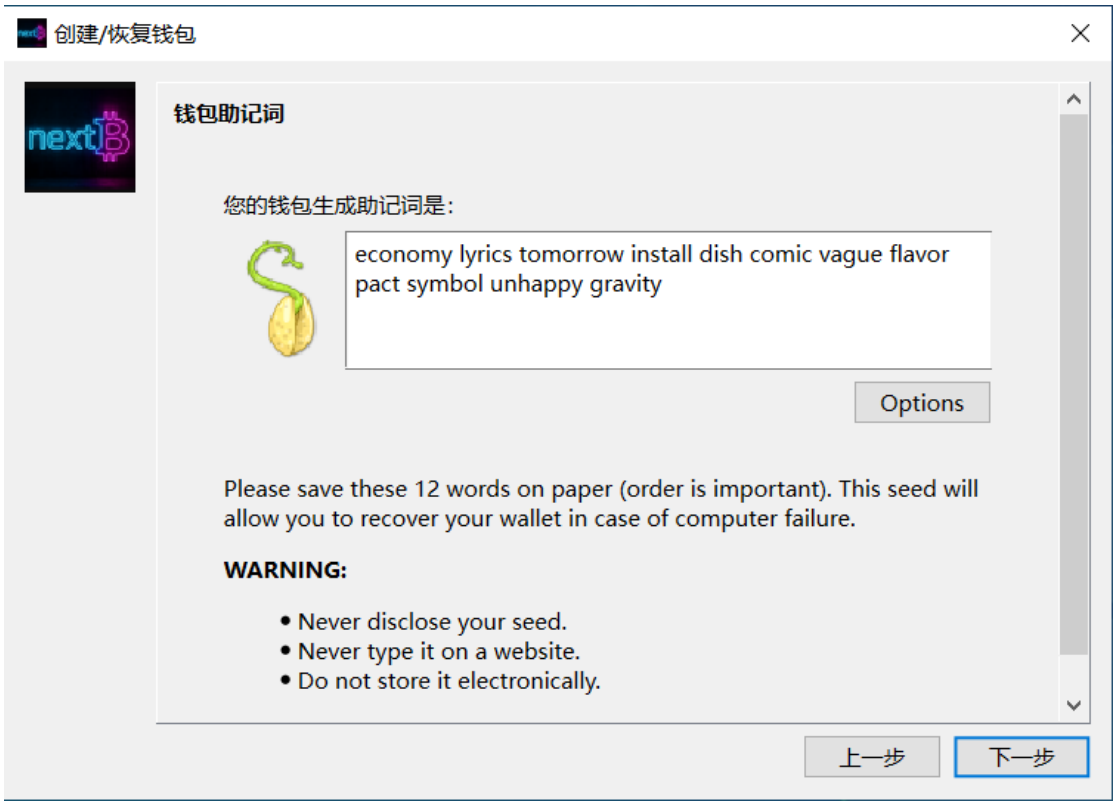


保持默认钱包名称，点击“下一步”。



选择“标准钱包”，然后选择“创建一个新的助记词”。





1、到此，我们获得了一套随机产生的 12 个单词的助记词。这就是我们算法中的**“脑口令 A”**。

economy lyrics tomorrow install dish comic vague flavor pact symbol unhappy gravity

1、 请注意： 如果我们继续点击“下一步”，钱包软件会直接使用这套助记词生成一个普通的比特币钱包。但这不是我们的目的。我们的核心安全策略是，**这套助记词本身对应的地址里，永远不会存放大量资产。**我们只是利用它作为生成我们高级脑钱包的一个公开输入。因此，现在可以直接关闭 NextBTC Wallet。

2、 运行我们准备好的比特币脑钱包生成器。

Passphrase / 脑口令： 填入刚才生成的那 12 个助记词。

Salt / 盐值： 这里我使用一个只有我自己知道其含义的字符串：“达哥的 QQ 号和微信账号和网站地址用‘达哥’连接成的字符串”：2082007308 达哥 btcdage2011 达哥 btcdage.rf.gd。**这是绝对核心的秘密，绝不外泄。**

Hash Times / 哈希次数： 我选择 2011，以纪念达哥第一次挖矿的年份。**这个数字也是核心秘密。**

Generate Count / 生成数量： 暂时选 1，然后点击“开始计算”。



将参数填入脑钱包生成器。



1、 工具为我们生成了一套包含 5 行数据的密钥对。分别是：私钥、公钥、传统地址（1 开头）、隔离见证私钥（p2wpkh 格式）和原生隔离见证地址（bc1 开头）。在当前比特币网络环境下，我们只需要关注最后两行：**bc1 开头的囤币地址和对应的私钥。**
我们这次生成的地址和私钥是：
私钥:

p2wpkh:KzFrb8x5g6RvEHKT6hmDwcgjzVcbk5LqayQDzh4YHqDVuKi291Nz

地址:

bc1q36te9kcenqyv4jhzh5kfrytlyrf3n3f2jvrrpf

2、 这个工具的强大之处在于，只要“脑口令”、“盐值”和“哈希次数”这三个参数固定，你可以生成海量（最多 1000 个）的、不同但有确定性顺序的地址。这意味着你可以轻松管理大量的囤币地址，而无需备份每一个私钥。



将生成数量改为 10， 可以看到工具生成了 10 套对应的地址和私钥。

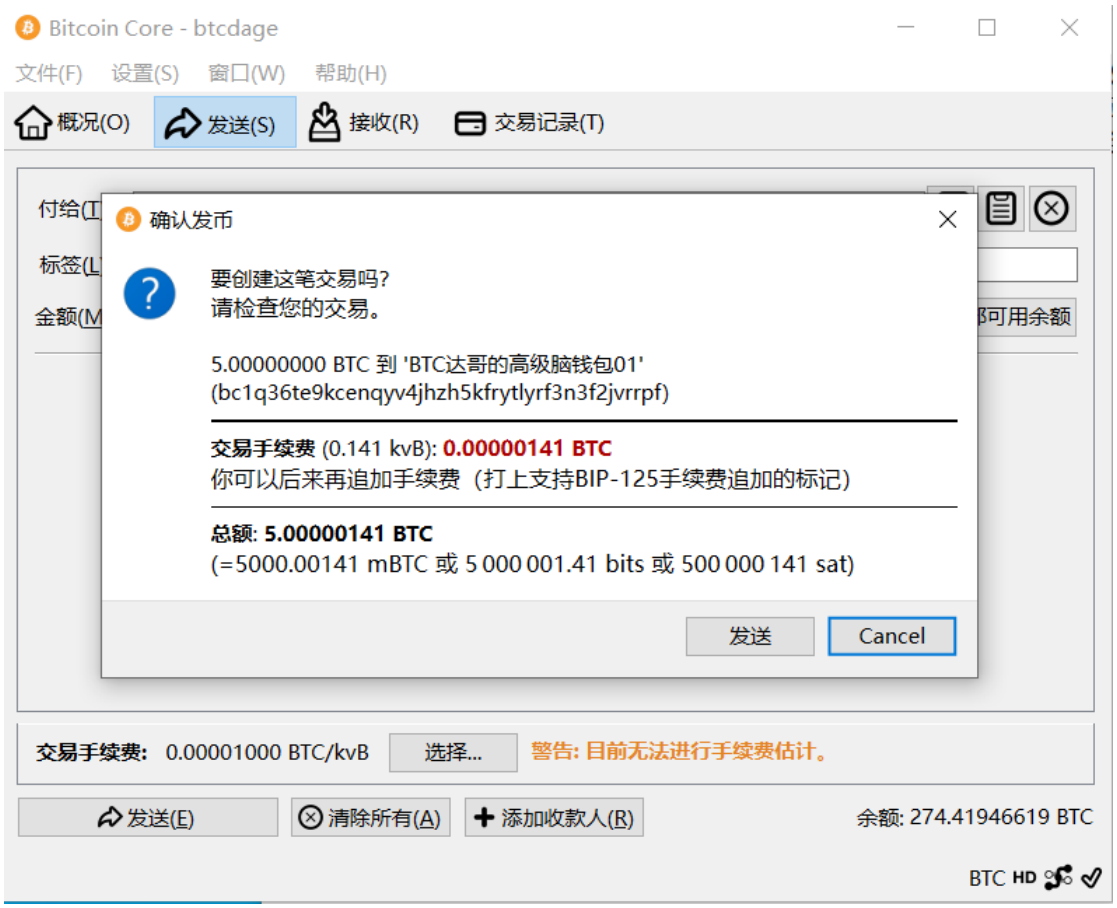
1、核心安全原则： 因为算法和参数是固定的，无论你重复运算多少次，得到的结果都完全一致。因此，在日常操作中，我们**完全不需要记录或备份任何私钥**。我们只需要记住脑口令、盐值和哈希次数。当需要接收比特币时，离线运行该工具生成地址即可；当需要花费比特币时，同样离线运行该工具，得到私钥后导入轻钱包进行签名转账。

2、下面我们开始测试。首先，我们将一笔资金（例如 5 BTC）转入前面生成的第一个高级脑地址： bc1q36te9kcenqyv4jhzh5kfrytlyrf3n3f2jvrrpf。这个过程可以模拟从交易所提币到你的冷地址。

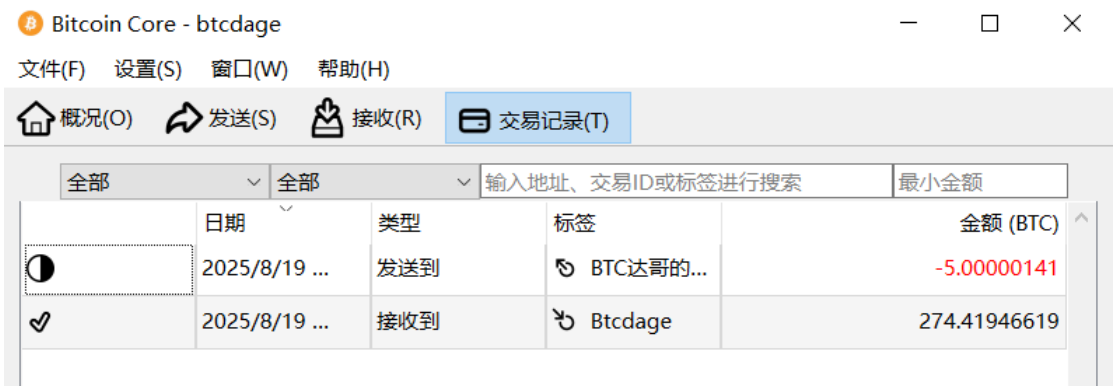
。

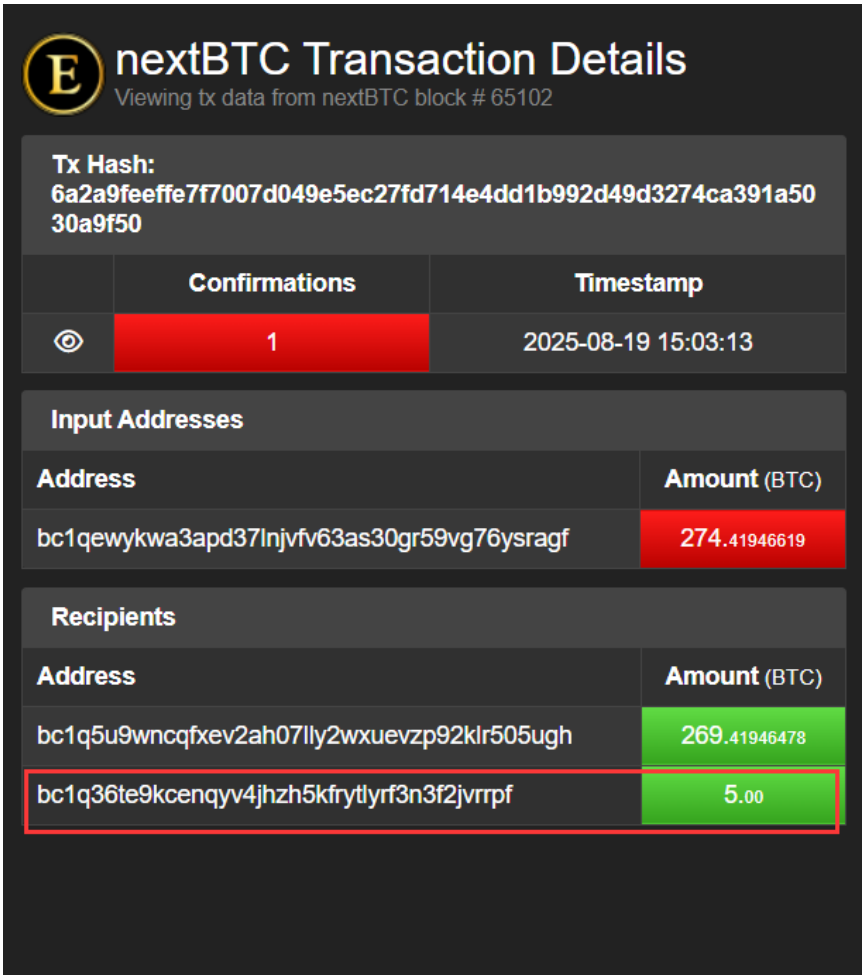


从一个全节点钱包发送 5 BTC 到我们生成的高级脑地址。


Bitcoin Core - btcdage
文件(F) 设置(S) 窗口(W) 帮助(H)
概况(O) 发送(S) 接收(R) 交易记录(T)
确认发币
要创建这笔交易吗?
请检查您的交易。
5.00000000 BTC 到 'BTC达哥的高级脑钱包01'
(bc1q36te9kcenqyv4jhzh5kfrytlyrf3n3f2jvrrpf)
交易手续费 (0.141 kvB): 0.00000141 BTC
你可以后来再追加手续费（打上支持BIP-125手续费追加的标记）
总额: 5.00000141 BTC
(=5000.00141 mBTC 或 5 000 001.41 bits 或 500 000 141 sat)
发送
Cancel
交易手续费: 0.00001000 BTC/kvB
选择...
警告: 目前无法进行手续费估计。
发送(E)
清除所有(A)
添加收款人(R)
余额: 274.41946619 BTC
BTC HD


Bitcoin Core (GUI node for Bitcoin)
Bitcoin Core - 送出交易
日期: 2025/8/19 14:59
金额: -5.00000141 BTC
类型: 发送到
标签: BTC达哥的高级脑钱包01

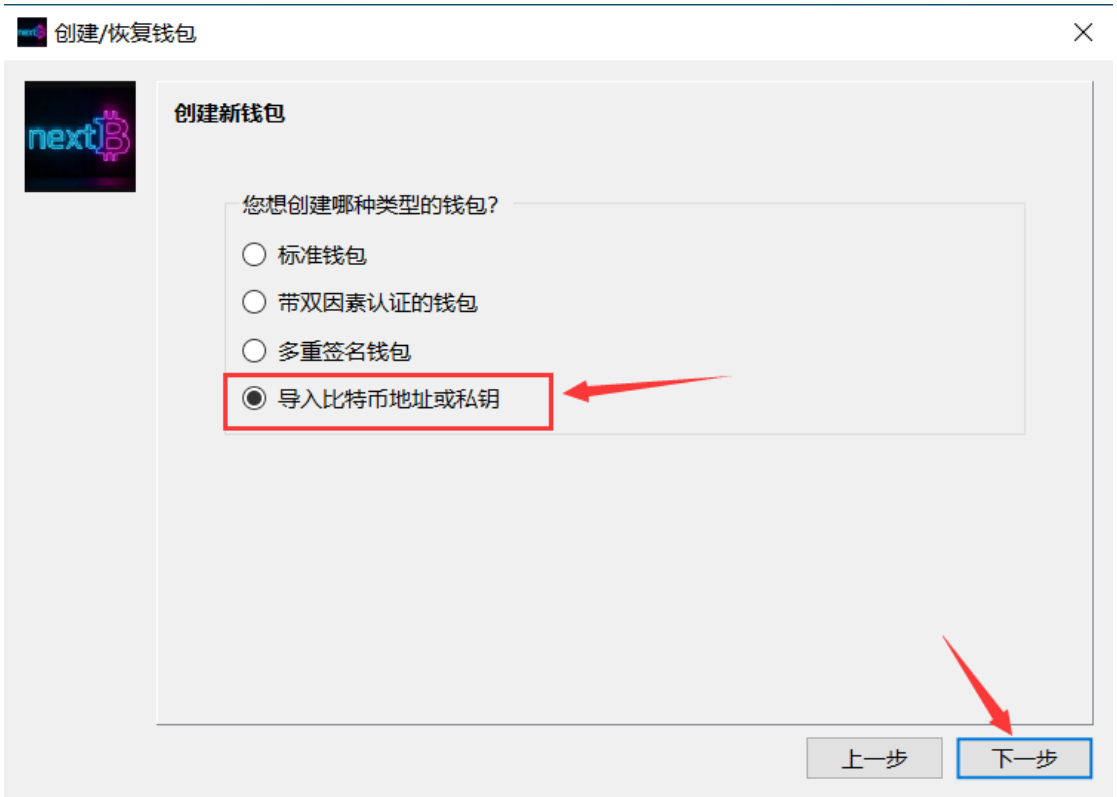

Bitcoin Core - btcdage
文件(F) 设置(S) 窗口(W) 帮助(H)
概况(O) 发送(S) 接收(R) 交易记录(T)
全部
全部
输入地址、交易ID或标签进行搜索
最小金额
日期
类型
标签
金额 (BTC)
2025/8/19 ...
发送到
BTC达哥的...
-5.00000141
2025/8/19 ...
接收到
Btcdage
274.41946619



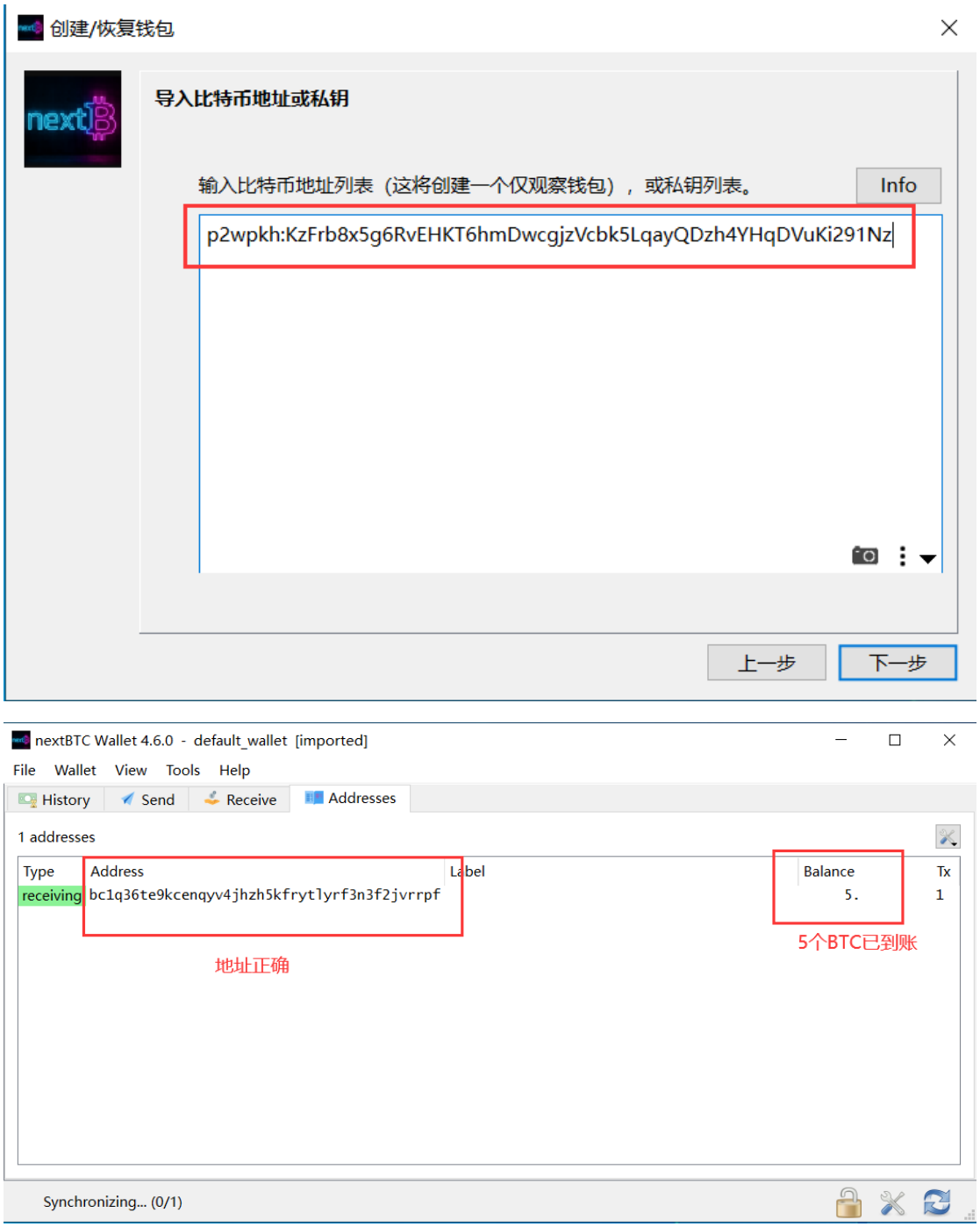
交易已发送，在区块浏览器上可以看到这笔交易正在确认中。

1、好了，我们从区块浏览器上已经看到 5 个 BTC 已经到账。现在，我们来验证如何花费这笔钱。我们再次打开 NextBTC Wallet 轻钱包，这次选择导入我们刚刚在离线环境中生成的私钥，看看资产是否能够成功识别。





在钱包创建流程中，选择“导入比特币地址或私钥”。



将我们生成的 p2wpkh 格式的私钥粘贴进去，钱包成功识别出地址和余额，证明私钥是正确的。

3、 到此，囤币地址的收款和私钥验证已测试成功。下面，我们把这 5 个币发送给批量生成的第二个囤币地址，以模拟一次完整的资金操作。

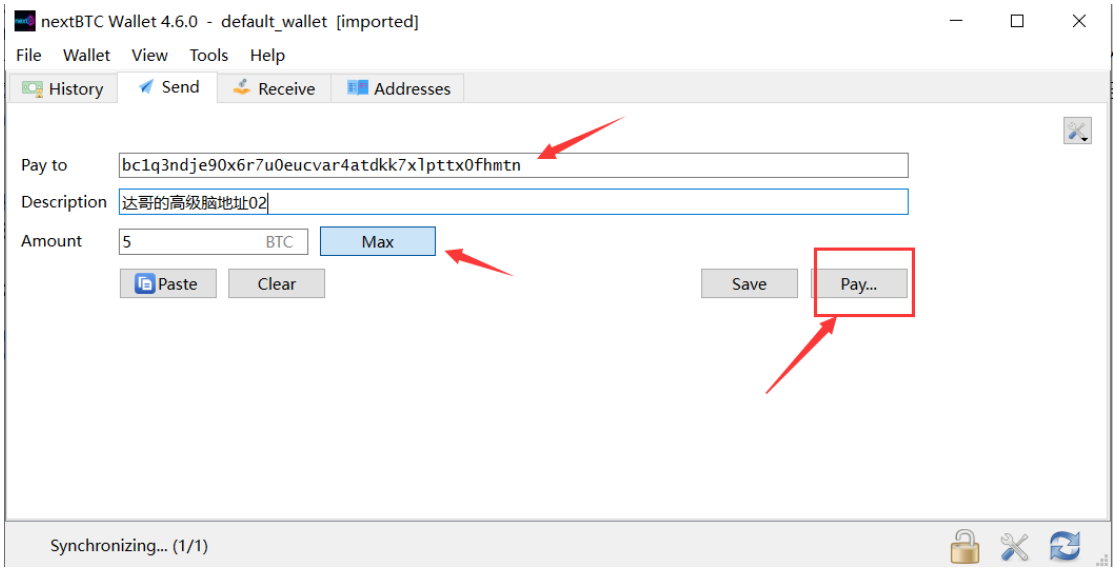
第二个地址是：

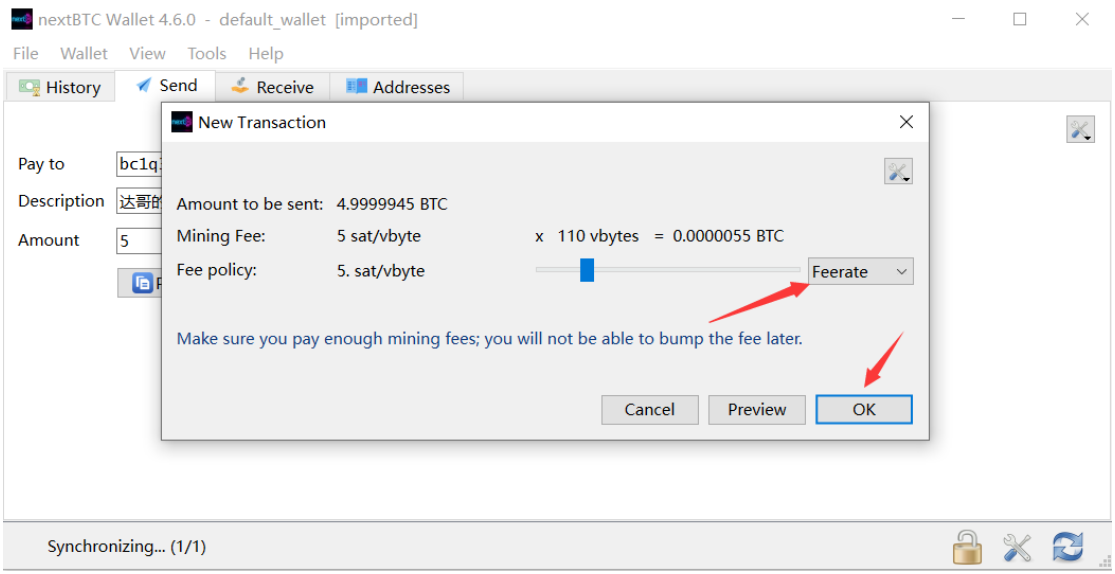
bc1q3ndje90x6r7u0eucvar4atdkk7xlpttx0fhmtn

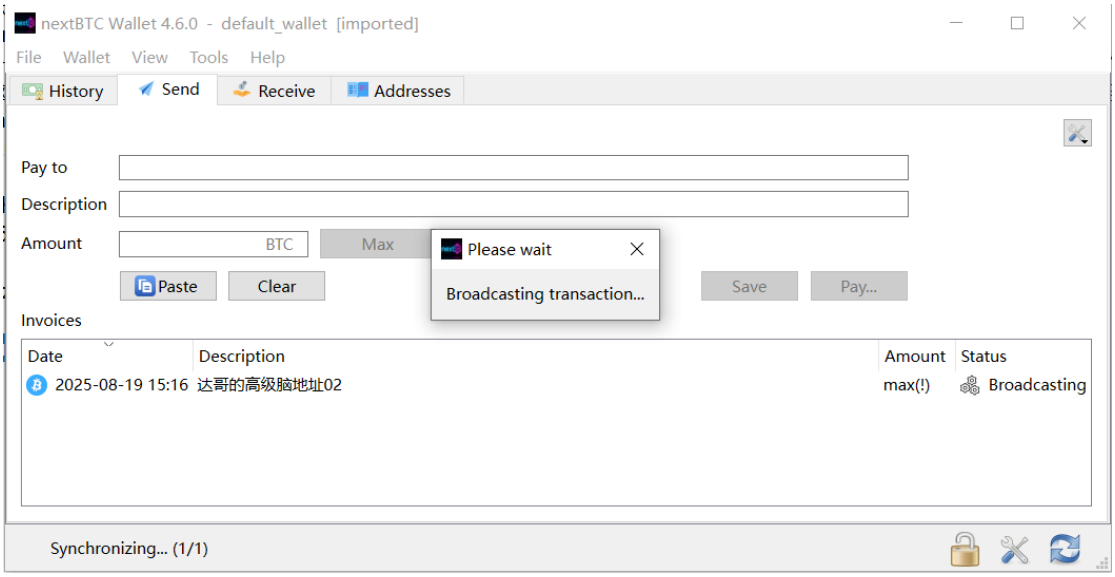


在脑钱包生成器中找到第 10 行的地址（即第二个地址）。

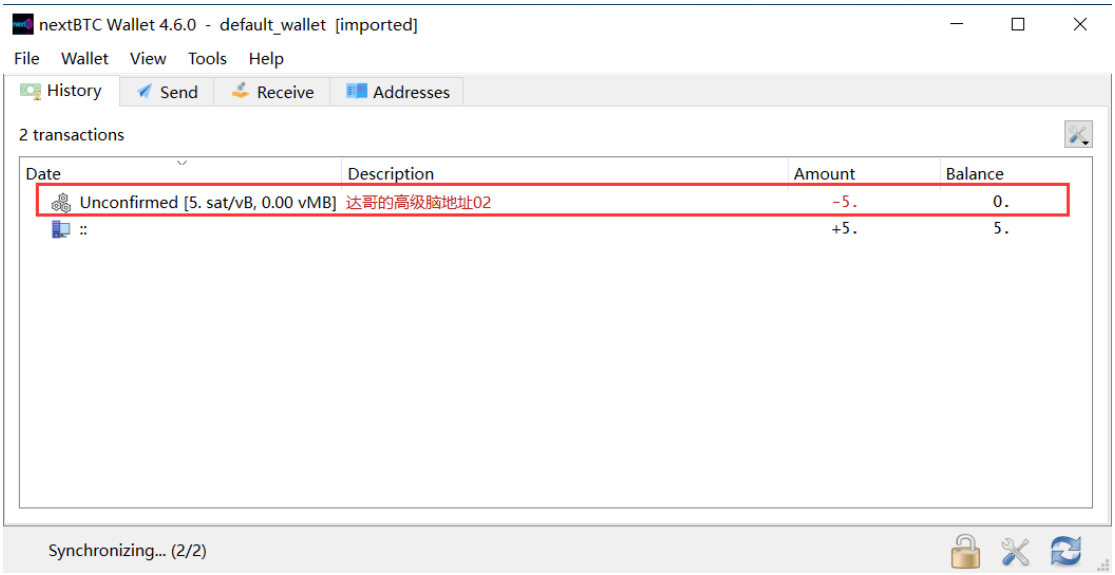
1、在轻钱包中选择“发送”，将第二个高级脑地址填入。选择“全部转出（MAX)”，然后点击支付（Pay)。

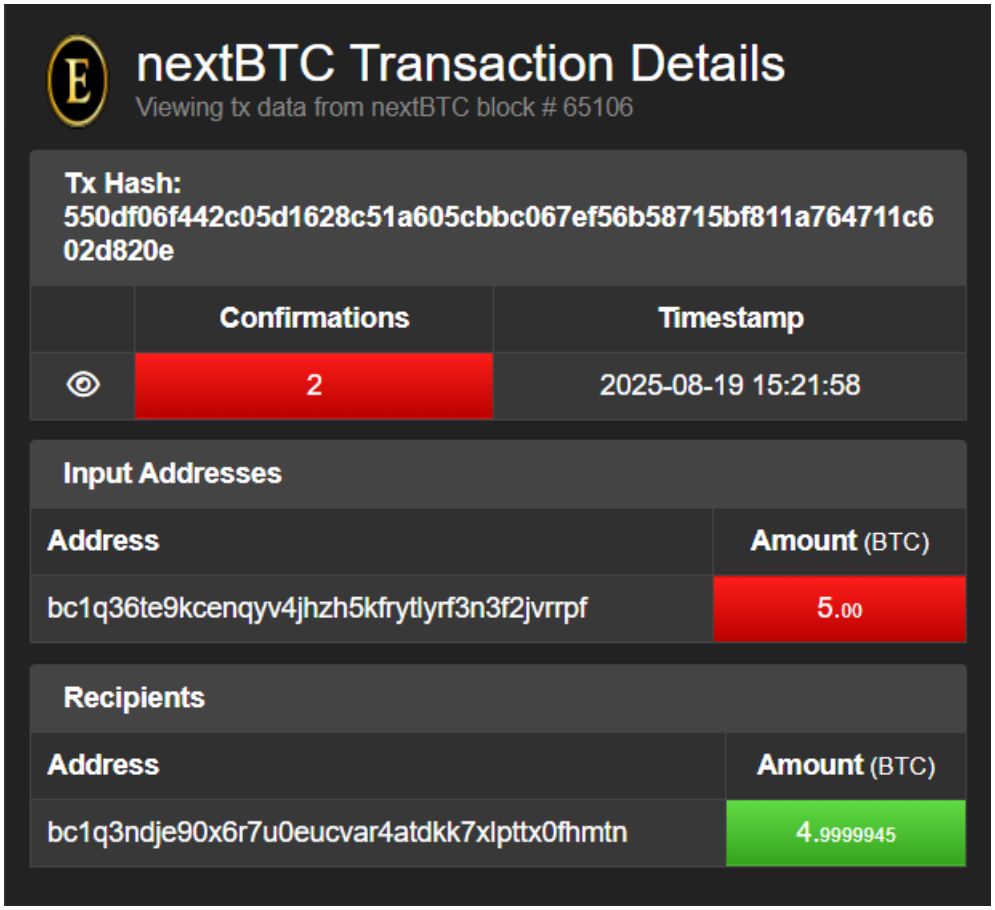


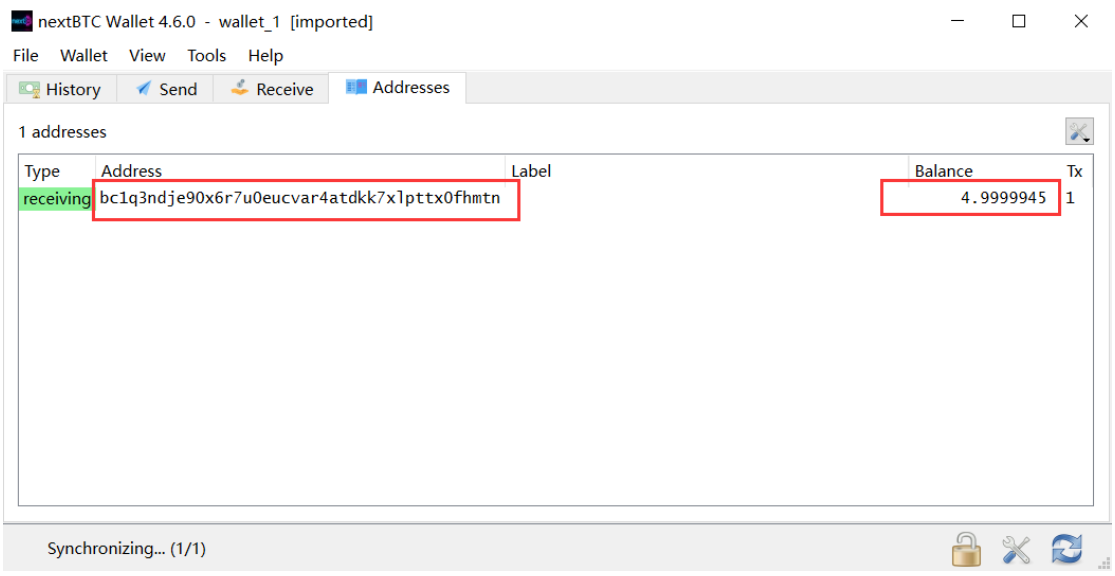

nextBTC Wallet 4.6.0 - default_wallet [imported]
File Wallet View Tools Help
History Send Receive Addresses
New Transaction
Amount to be sent: 4.9999945 BTC
Mining Fee: 5 sat/vbyte x 110 vbytes = 0.0000055 BTC
Fee policy: 5. sat/vbyte
Feerate
Make sure you pay enough mining fees; you will not be able to bump the fee later.
Cancel Preview OK
Pay to
Description
Amount
Synchronizing... (1/1)


nextBTC Wallet 4.6.0 - default_wallet [imported]
File Wallet View Tools Help
History Send Receive Addresses
Pay to
Description
Amount
BTC
Max
Please wait
Broadcasting transaction...
Paste Clear
Save Pay...
Invoices
Date Description Amount Status
2025-08-19 15:16 达哥的高级脑地址02 max(!) Broadcasting
Synchronizing... (1/1)

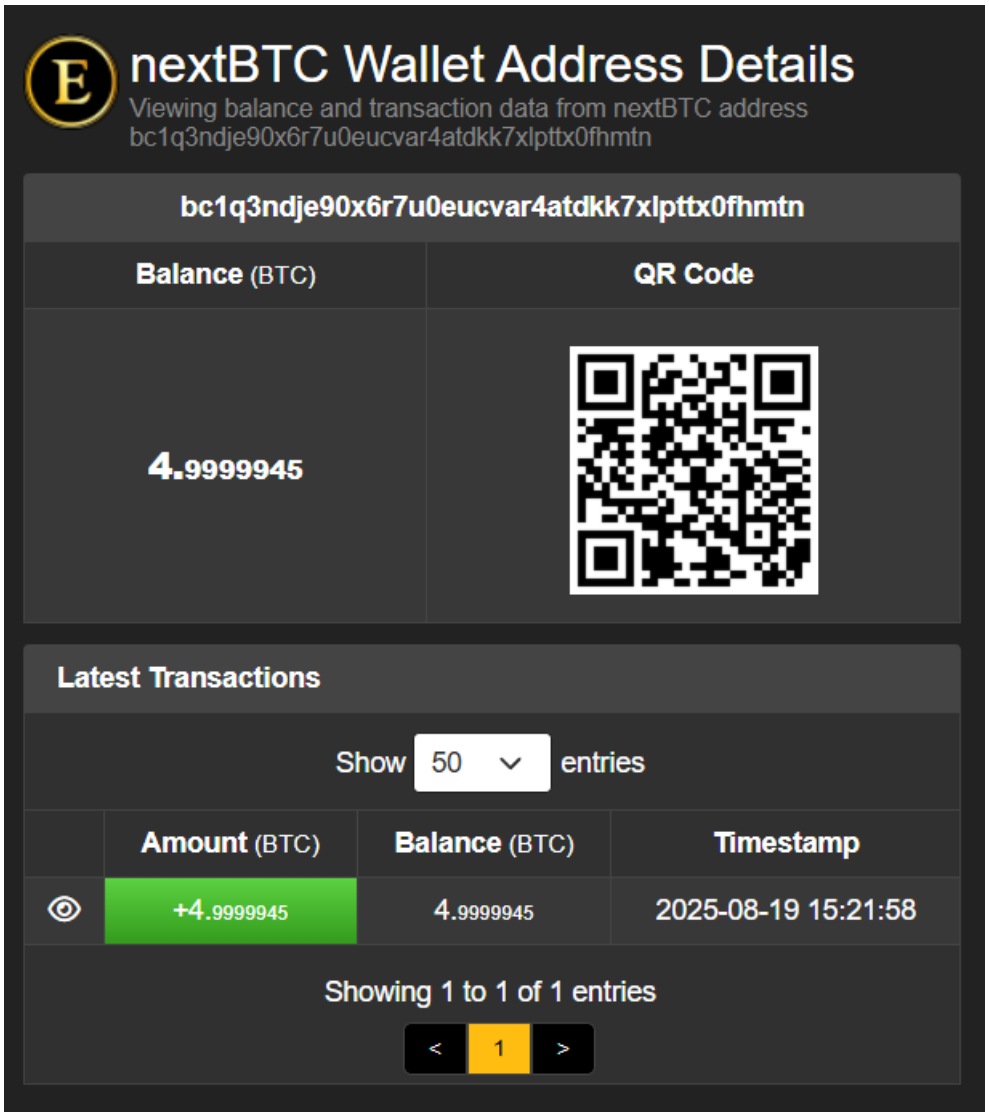
钱包显示交易正在广播。


nextBTC Wallet 4.6.0 - default_wallet [imported]
File Wallet View Tools Help
History Send Receive Addresses
2 transactions
Date Description Amount Balance
Unconfirmed [5. sat/vB, 0.00 vMB] 达哥的高级脑地址02 -5. 0.
:: +5. 5.
Synchronizing... (2/2)





交易被打包确认后，在区块浏览器和钱包中都能看到转账成功。



在区块浏览器中查看第二个地址，确认资金已到账。

四、 核心安全与隐私：

高级脑钱包的安全性基石在于其独特的设计，它将“记忆”和“物理备份”巧妙地分离开来：

脑口令 A (12 个助记词):

它可以、也应该进行多重物理备份，存放在不同的安全地方。你不用担心它被盗，因为这个助记词本身对应的钱包地址里没有任何资产（或者只放少量资产作为“哨兵钱包”来监控异常）。它的作用仅仅是算法的一个公开部分。

盐值 B 和哈希次数 N:

这两者是你的终极秘密，绝对不能记录在任何物理世界或网络空间，只能牢牢记在你的脑中。盐值 B 可以是你生命中一段独特的、足够长且无规律的记忆，对自己来说意义非凡，但对别人来说毫无意义。

这种模式的安全性极高，因为攻击者即使拿到了你的助记词备份，也无法推算出你的真实资产地址。**真正的私钥只在你需要转账时，通过离线工具在内存中瞬间生成，用完即焚，不留任何痕迹。**

五、 关于 nextBTC

我们这次实战演示选择在一个名为：NextBTC（nextbtc2025/nextbtc103）的项目中进行，它是一个比特币的平行链项目。选择它的原因在于，它为我们提供了一个理想的、几乎零风险的实战演练环境。

NextBTC 在技术特性上和比特币几乎完全兼容，同时又拥有一个独立且完整的生态系统，包括：

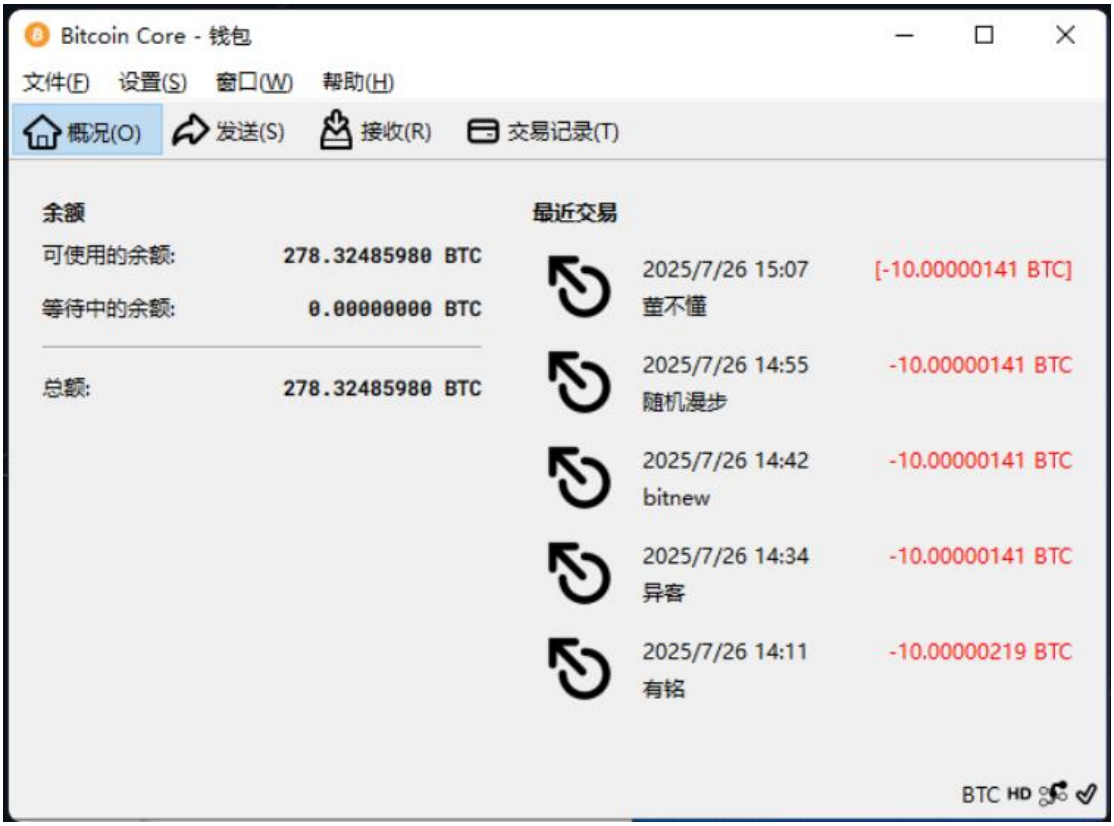
★全节点钱包软件

★POW 挖矿机制（支持早期 CPU 挖矿体验）

★完善的区块链浏览器

★轻钱包软件

对于任何想要深入学习比特币、测试钱包安全方案或体验挖矿过程的用户来说，NextBTC 都是一个绝佳的测试链。它让我们可以在一个真实的环境中自由测试，而无需担心造成任何实际的资产损失。



这次高级脑钱包的演示，完整覆盖了比特币转账、私钥导入等关键功能，都可以在 nextBTC 这条测试链上安全地进行。

重要提示： 测试终归是测试。请不要将本教程中用于测试的算法和参数直接用于存储你的实际资产。当你真正准备生成自己的囤币地址时，务必在完全离线的计算机上进行所有操作，并且在签名转账之前，私钥绝不可以暴露在任何联网的设备中。

为了让 nextBTC 这条测试链能够持续为大家服务，我们欢迎有兴趣的朋友部署全节点或参与挖矿（支持全节点 solo 和 PPLNS 矿池）。在其官网 https://nextbtc.rf.gd/ 中可以找到完整的图文教程。

以下是 nextBTC 的一些技术参数：

特性	参数
创世区块信息	"I COME, I SEE, I HODL."
算法	SHA256d
区块时间	2.5 分钟 (150 秒) (交易确认速度提升 4 倍)
初始区块奖励	3.125 nextBTC
奖励减半周期	每 800,000 个区块
总供应量	5,000,000 nextBTC (稀缺度提升超过 4 倍)
地址格式	Bech32 (bc1...), Base58
SegWit 激活	自创世区块起 (高度 0)
RPC 端口	8554
P2P 端口	8666