





















排名靠前的几个币种d

参考网站

1	 Bitcoin	\$142,521,649,676	\$8,301.11	\$7,434,270,000	17,168,987 BTC	7.31%		...
2	 Ethereum	\$48,138,614,509	\$477.09	\$2,320,770,000	100,900,701 ETH	5.49%		...
3	 XRP	\$18,031,391,228	\$0.458631	\$332,366,000	39,315,683,476 XRP *	3.22%		...
4	 Bitcoin Cash	\$14,929,042,401	\$865.17	\$934,166,000	17,255,538 BCH	10.27%		...
5	 EOS	\$7,900,086,502	\$8.82	\$1,092,090,000	896,149,492 EOS *	11.46%		...
6	 Stellar	\$5,621,575,548	\$0.299541	\$106,661,000	18,767,299,129 XLM *	5.08%		...
7	 Litecoin	\$5,065,064,432	\$88.01	\$441,896,000	57,553,757 LTC	6.57%		...
8	 Cardano	\$4,502,650,632	\$0.173666	\$191,238,000	25,927,070,538 ADA *	5.29%		...
9	 IOTA	\$2,736,592,099	\$0.984552	\$41,830,600	2,779,530,283 MIOTA *	5.45%		...
10	 TRON	\$2,522,695,871	\$0.038369	\$340,246,000	65,748,111,645 TRX *	13.29%		...

瑞波币XRP

浏览器

XRP Charts

市场

OfferCreate

rs9tBkt96q9gwrePKPqimUuF7vErgMaker

buy 1,000.000001 XRP @ 17.484 BTC/XRP

OfferCreate

rETx8GBIH6fxhTctfHM9fGeyShqxozyD3xe

sell 1,528.450369 XRP @ 3.1383 XRP/CNY

OfferCreate

rETx8GBIH6fxhTctfHM9fGeyShqxozyD3xe

sell 1,528.450369 XRP @ 3.1103 XRP/CNY

OfferCreate

rENDnFwR3CPvrsPJd9XXeqVoXeVt2CpPWx

buy 6,600 XRP @ 17.737 BTC/XRP

OfferCreate

rENDnFwR3CPvrsPJd9XXeqVoXeVt2CpPWx

sell 0.367026 BTC @ 17.982 BTC/XRP

OfferCreate

rENDnFwR3CPvrsPJd9XXeqVoXeVt2CpPWx

buy 6,600 XRP @ 1,023.2 ETH/XRP

OfferCreate

rENDnFwR3CPvrsPJd9XXeqVoXeVt2CpPWx

sell 6.36504 ETH @ 1,036.9 ETH/XRP

OfferCancel

rENDnFwR3CPvrsPJd9XXeqVoXeVt2CpPWx

Offer 22054042 cancelled

OfferCancel

rENDnFwR3CPvrsPJd9XXeqVoXeVt2CpPWx

Offer 22053980 cancelled

OfferCancel

rENDnFwR3CPvrsPJd9XXeqVoXeVt2CpPWx

Offer 22054041 cancelled

OfferCancel

rENDnFwR3CPvrsPJd9XXeqVoXeVt2CpPWx

Offer 22053979 cancelled

2B66BCB994297D3248B6359C2E08BDB5052BD84197274ACD3885CDE5850F2A03

去

描述

生

状态:

这笔交易是成功的,在分类帐和验证 **40352791** 在 **2018年7月26日,2:07点**。

描述:

这是一个 **OfferCreate** 事务。
rhS2H7ETM3wBkFETvYycoUm9FEDYi44Pg4 提供支付 **3234年 .908604 XRP** 为了获得 **10067年** 比赛 元。RippleChina (razqQKzjRdB4UxFPWf5NEpEG3WMkmgcXA)。
这个报价的汇率 **3.112 XRP /元**。
事务也将取消 rhS2H7ETM3wBkFETvYycoUm9FEDYi44Pg4 现有提供# **17647366**
事务的序号 **17647367**

备忘录:

事务没有备忘录。

交易成本:

发送这个事务消耗 **0.000027 XRP**。

国旗:

事务指定下列标志:

- tfFullyCanonicalSig**

总价值排名第三。总发行量1000亿（三位创始人分得200亿，拉森获得了95亿XRP800亿归公司所有）。特点

- 没有挖矿的过程。1000亿已经一开始就订好了

- 交易确认时间平均4秒
- 交易手续费几乎为0.手续费不给任何一个人，因为没有矿工。而是凭空消失，所以总量是一直减少的

主要用途是用来跨国转账和法币兑换。外汇交易中间依赖大量的中间方，整体效率非常低下、费用高昂，通常需要3-5天的到账时间，20-70刀的费用

而瑞波币可以做到4秒确认到账。其已经与数十家大型银行、金融机构开展了合作。

瑞波币原理

三种交易模式xCurrent、xRapid、xVia

xCurrent是由中间银行作为中转完成交易，xRapid是用XRP完成中间的交易，而xVia则是由网关作为中转完成交易

xCurrent，主要为银行与银行之间提供跨境交易。Ripple网络在银行间设立了分布式的账本，每当有银行A向银行B转账，可以靠中间银行C进行清算。实质上通过分布式账本，使A在C开设的银行账户及B在C开设的银行账户内的金额发生了转变。这种模式优点是速度较快、费用低，缺点是三家银行都需要加入Ripple网络，使用同一套分布式账本

xRapid，模式为支付方先将支付金额换成xrp，发送给收款方的银行，银行将收到的xrp转换成对应的货币，再支付给对应的收款方。这种模式相比xCurrent更灵活，只需要收款方的银行可以接受xrp并换成对应法币即可。当然将xrp转换为当地货币的步骤也可以由收款人自主完成。

xVia，是引入了网关的概念。网关就是Ripple系统的一个中介机构（类似银行），支付方可以先将任意货币先转给网关，再由网关将货币转换成其它货币，支付给收款人即可。这种模式最为灵活，支付方和收款方都不需要加入Ripple网络，只需要信任网关即可。

无论哪种方式，中间都依赖中心化机构（例如银行、网关等），才能完成整个环节。但是中心化机构就意味着安全性问题，尤其是网关，有可能会破产或者捐款逃跑。

Ripple目前的记账节点非常中心化，很大一部分的记账节点实际上是由Ripple自己控制的。

交易共识

瑞波网络有一个固定的信任节点列表，由列表的节点进行出块，超过80%的确认数就更新区块链。基本三秒一个块，tps1500

比特币现金 bitcoin cash

bch是于17年8月1日，区块高度478598硬分叉完成，按照比特币1：1分发，总量2100万。

删除隔离见证、区块上限升级为8M，坚持链上扩容，解决了旧版比特币系统中手续费高、确认慢、实用性差等问题，履行最初的比特币作为「点对点电子现金」的承诺。目前比特币现金由8个不同的开发团队维护，市值曾达到第二名，

今年五月份硬分叉32m一个块

[大块](#)

[大交易](#)

EOS

特点

- eos可以处理百万级别的交易
- 共识算法（dpos），每3秒产生一个块，并且只有一个生产者被授权在任何给定的时间点产生一个块。如果在预定时间没有产生该块，则该时间的块被跳过。当一个或多个区块被跳过时，区块链中有6个或更多秒的空

DPOS共识

原理是让每一个持有比特股的人进行投票，由此产生101位代表，我们可以将其理解为101个超级节点或者矿池，而这101个超级节点彼此的权利是完全相等的。从某种角度来看，DPOS有点像是议会制度或人民代表大会制度。如果代表不能履行他们的职责（当轮到他们时，没能生成区块），他们会被除名，网络会选出新的超级节点来取代他们。DPOS的出现最主要还是因为矿机的产生，大量的算力在不了解也不关心比特币的人身上，类似演唱会的黄牛，大量囤票而丝毫不关心演唱会的内容

[浏览器](#)

stellar 恒星币 xml

星和瑞波是亲兄弟是同一创始人，恒星是瑞波的升级版

总量1000亿，其中95%将通过免费发放的形式提供给用户

恒星支付网络以恒星币为基础货币，用户能够通过其转账任意一种货币，包括美元、欧元、人民币、日元或者比特币，简便易行快捷，交易确认在几秒以内完成。50%通过直接分发计划分配给全世界，25%通过增加覆盖计划分配给非营利组织以给予金融服务匮乏的人群，20%通过比特币计划分配，5%留作运营费用恒星币运营。

币圈著名漏洞

The DAO漏洞

the DAO：DAO 是Decentralized Autonomous Organization（分布式自治组织）的简称，the DAO是一个基于以太坊区块链平台的迄今为止世界上最大的众筹项目。其目的是让持有The DAO代币的参与者通过投票的方式共同决定被投资项目，整个社区完全自制，并且通过代码编写的智能合约来实现。于2016年5月28日完成众筹，共募集1150万以太币，在当时的价值达到1.49亿美元。

6月17日，加密货币和区块链社区发生了一次大地震，the DAO 被黑客攻击了。价值6千万美元的以太币被盗！

[合约地址](#)

```

function splitDAO(uint _proposalID, address _newCurator) noEther onlyTokenholders returns (bool
_success) {
    // ...
    // XXXXX Move ether and assign new Tokens. Notice how this is done first!
    uint fundsToBeMoved = (balances[msg.sender] * p.splitData[0].splitBalance) /
p.splitData[0].totalSupply;
    if (p.splitData[0].newDAO.createTokenProxy.value(fundsToBeMoved)(msg.sender) == false)
        // XXXXX This is the line the attacker wants to run more than once
        throw;
    // ...
    // Burn DAO Tokens
    Transfer(msg.sender, 0, balances[msg.sender]);
    withdrawRewardFor(msg.sender); // be nice, and get his rewards
    // XXXXX Notice the preceding line is critically before the next few
    totalSupply -= balances[msg.sender]; // XXXXX AND THIS IS DONE LAST
    balances[msg.sender] = 0; // XXXXX AND THIS IS DONE LAST TOO
    paidOut[msg.sender] = 0;
    return true;
}

function withdrawRewardFor(address _account) noEther internal returns(bool _success) {
    if ((balanceOf(_account) * rewardAccount.accumulatedInput()) / totalSupply <
paidOut[_account])
        throw;
    uint reward = (balanceOf(_account) * rewardAccount.accumulatedInput()) / totalSupply -
paidOut[_account];
    if (!rewardAccount.payOut(_account, reward)) // XXXXX vulnerable
        throw;
    paidOut[_account] += reward;
    return true;
}

function payOut(address _recipient, uint _amount) returns (bool) {
    if (msg.sender != owner || msg.value > 0 || (payOwnerOnly && _recipient != owner))
        throw;
    if (_recipient.call.value(_amount)()) { // XXXXX vulnerable
        PayOut(_recipient, _amount);
        return true;
    } else {
        return false;
    }
}

```

先看下面两段代码：

```
address addr = 0x6c8f2a135f6ed072de4503bd7c4999a1a17f824b; if(!addr.call.value(20 ether)()){
throw; }
```

以及：

```
address addr = 0x6c8f2a135f6ed072de4503bd7c4999a1a17f824b; if(!addr.send(20 ether)){ throw; }
```

这两段代码都是向0x6c8f...的合约地址发送20个ether，第二段代码没有漏洞，而第一段代码却存在严重的安全漏洞。为什么？

我们先来看一下`addr.call.value()`（注意：是两个括号，第一个括号是对要转移多少以太币的赋值，第二个括号是方法的调用）和`addr.send()`的区别。两者都是向某个地址发送以太币，都是一个新的message call，不同的是这两个调用的gaslimit不一样。`send()`给予0的gas（相当于`call.gas(0).value()`），而`call.value()`给予全部（当前剩余）的gas。

注：对于需要调用fallback函数又没有给予任何gas的情况，EVM将自动把gas调整为不超过2300。

```
splitDao
  withdrawRewardFor （第一次调用）
    payOut
      recipient.call.value>()
        splitDao
          withdrawRewardFor （第二次调用）
            payOut
              recipient.call.value>()
```