grincon0
18.11.09 // c-base berlin

Cross-chain atomic swaps with Grin

Jasper van der Maarel

Introduction

Grin is fast approaching mainnet!

An economy will form around Grin

A need to change between Grin and other cryptocurrencies

Variety of exchanges, with different technologies

Centralized

Decentralized

Centralized

Decentralized

Central exchanges

BINANCE kraken

BITTREX POLONIEX

Centralized

Decentralized

Central exchanges

BINANCE kraken

BITTREX POLONIEX

- Instant trades
- High liquidity, advanced options
- Surrender control of coins
- KYC
- Regulations / censorship

Centralized

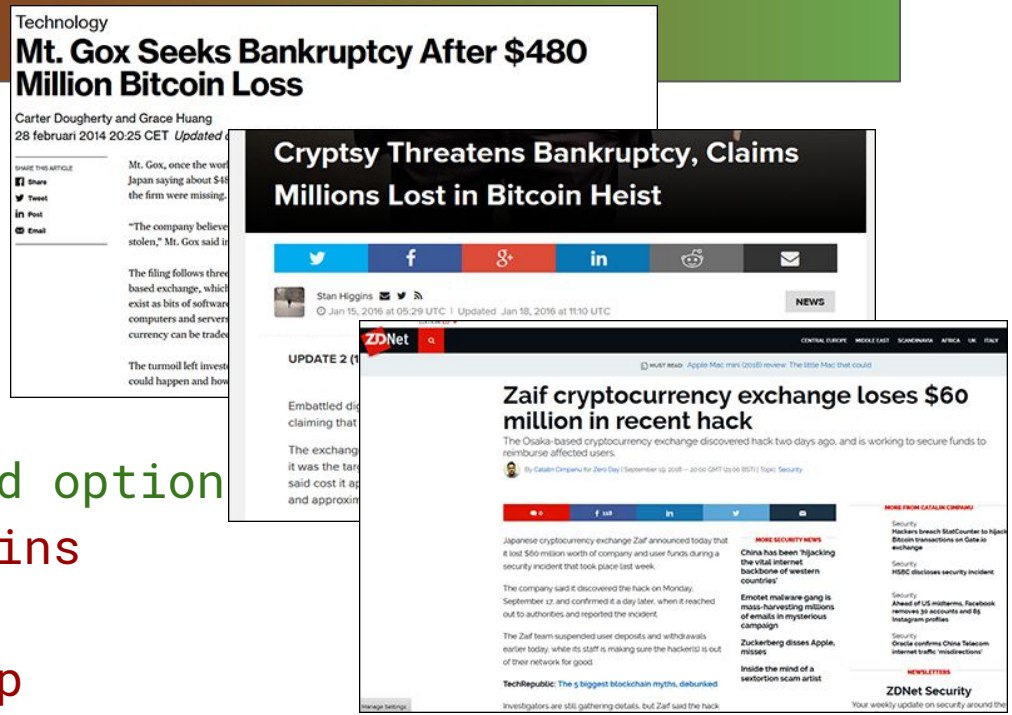
Decentralized

Central exchanges

BINANCE kraken

BITTREX POLONIEX

- Instant trades
- High liquidity, advanced option
- Surrender control of coins
- KYC
- Regulations / censorship



Centralized

Decentralized

Non-custodial exchanges

ShapeShift changelly

- Greater control of funds
- KYC
- Regulations

Centralized

Decentralized

On-chain decentralized exchanges

0x

EtherDelta

- Full control of funds
- “Anonymity”
- Trades are slow
- Poor scaling
- Single chain

Centralized

Decentralized

Cross-chain atomic swaps

- Full control of funds
- Can be fully p2p
- Anonymity
- Trades are slow

What is a cross-chain atomic swap?

Swap: trade without intermediary trusted party

Atomic: all-or-nothing. Either both parties complete the swap or neither of them does

Cross-chain: between currencies (here: Grin + BTC/ETH)

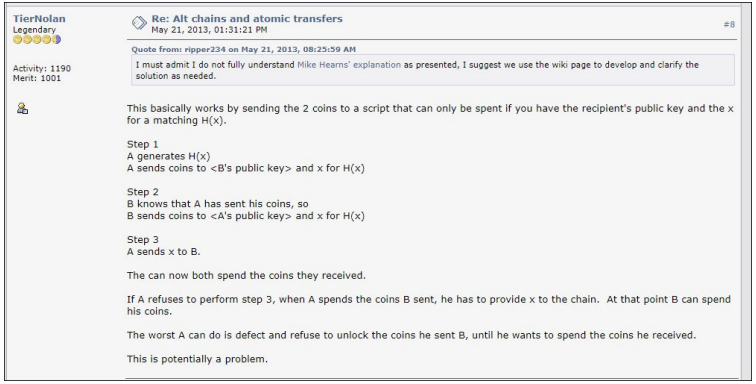
Atomic swap history

First proposal: 2013 by Tier Nolan on Bitcointalk

Weakness: no refunds

OP_CLTV soft fork

First BTC atomic swaps ~1y ago



HTLC atomic swaps

- Alice generates secret **q**
- Alice locks BTC with **H(q)**, unlockable by
  - Bob if he knows **q**
  - Alice if sufficient time has passed
- Bob locks altcoin with **H(q)**, unlockable by
  - Alice by revealing **q**
  - Bob if sufficient time has passed
- Relative time of the locks is important

HTLC atomic swaps

- Alice generates secret **q**
- Alice locks BTC with **H(q)**, unlockable by
  - Bob if he knows **q**
  - Alice if sufficient time has passed
- Bob locks altcoin with **H(q)**, unlockable by
  - Alice by revealing **q**
  - Bob if sufficient time has passed
- Relative time of the locks is important
- Requires timelocks
- Requires hash pre-images
- Both sides of swap are linkable (use the same hash)

Grin swaps: in theory

MW atomic swaps

High level overview:

1. Bob generates secret **q**. Locks BTC/ETH in a 2-of-2 multisig using Alice’s key and **q**. Refund to Bob after 24h
2. Alice locks grins (with help of Bob) in a 2-of-2 multisig using Alice’s and Bob’s keys. Refund to Alice after 12h
3. Alice and Bob cooperate to send grins to Bob. When Bob finalizes the transaction, he automatically reveals **q**
4. Alice claims the BTC/ETH

1 - Bob locks BTC/ETH

Bitcoin: P2SH address

- Bob deposits agreed amount
- Output can be spent by
  - Alice, if she knows **q**
  - Bob, if 24 hours have passed

Ethereum: similar, but smart contract

```
OP_IF
  <now+24h>
  OP_CLTV
  OP_DROP
  <Bob's pubkey>
  OP_CHECKSIG
OP_ELSE
  OP_2
  <Alice's pubkey>
  <q pubkey>
  OP_2
  OP_CHECKMULTISIG
OP_ENDIF
```

2 - Alice (and Bob) lock grins

Remember: no scripts, interactive transactions

Outputs are Pedersen commitments

$$C = x*G + v*H$$

2 - Alice (and Bob) lock grins

How do we make a multisignature?

Create a new output

$$O = (x_A + x_B) * G + v * H$$

Complication: building rangeproof requires knowledge of blinding factor → replaced by interactive process

2 - Alice (and Bob) lock grins

Second complication: holding coins hostage

Once tx is final, neither party has full control of the output

If Bob refuses to cooperate, Alice’s coins will be stuck

Solution: build a refund back to Alice, before finalizing the multisignature

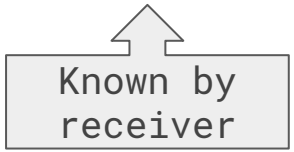
This transaction has a 12h timelock, it can not be mined in a block before its expiration

3 - Build tx to spend grins

In general, transactions have to balance out

For example, 1 input -> 2 output transaction:

$$O_R + O_{ch} - I_S = (x_R + x_{ch} - x_S) * G + 0 * H$$

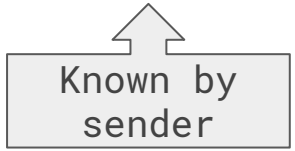


3 - Build tx to spend grins

In general, transactions have to balance out

For example, 1 input -> 2 output transaction:

$$O_R + O_{ch} - I_S = (x_R + x_{ch} - x_S) * G + 0 * H$$



3 - Build tx to spend grins

In general, transactions have to balance out

For example, 1 input -> 2 output transaction:

$$O_R + O_{ch} - I_S = (x_R+x_{ch}-x_S)*G + 0*H$$

Schnorr signature for excess: both parties can generate a partial signature for their own contribution to the tx

Total signature is sum of parts:

$$(s,k*G) = (s_S+s_R, k_S*G+k_R*G)$$

3 - Build tx to spend grins

In our case, Alice and Bob build a tx to spend the multisig

$$I = (x_A+x_B)*G + v*H$$

And create a new output solely controlled by Bob

Trick: adaptor signature

Bob will calculate his partial signature s_B, but instead send $s_{adap} = s_B + q$ to Alice

3 - Build tx to spend grins

Alice can cryptographically verify that she in fact did receive $\mathbf{s_B + q}$

She can now safely send $\mathbf{s_A}$ to Bob

Bob will publish the transaction with total signature $\mathbf{s = s_A + s_B}$ to the Grin network, claiming the coins

4 - Claim BTC/ETH

Alice monitors the Grin chain for the transaction

From this she learns the total signature **s**

She performs simple arithmetic to find **q**:

$$q = s_{adap} + s_A - s$$

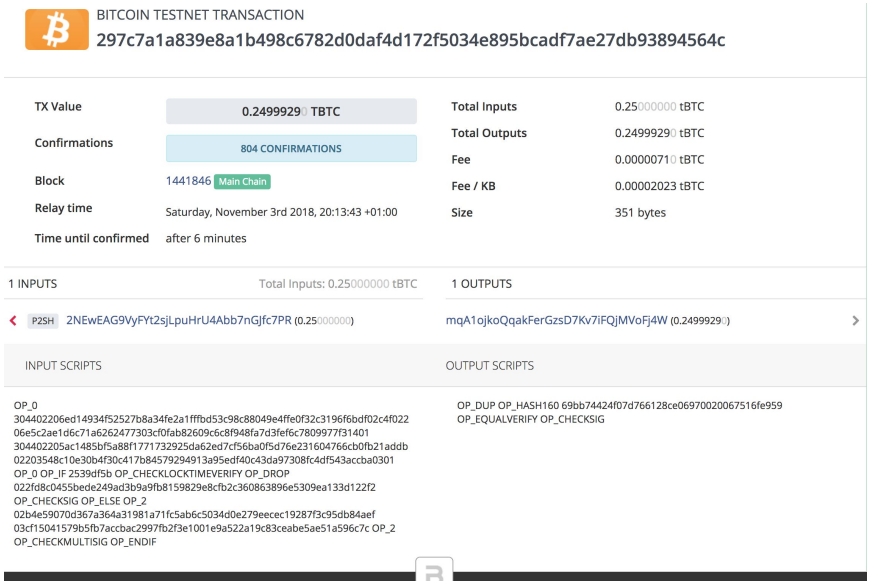
This allows her to spend the multisig on the BTC/ETH chain

The swap is now complete!

Grin swaps: in practice

Atomic swaps on testnet

T4: first Grin-tBTC swap



BITCOIN TESTNET TRANSACTION

297c7a1a839e8a1b498c6782d0daf4d172f5034e895bcadf7ae27db93894564c

TX Value	0.2499929 TBTC	Total Inputs	0.25000000 tBTC
Confirmations	804 CONFIRMATIONS	Total Outputs	0.24999290 tBTC
		Fee	0.00000710 tBTC
Block	1441846 Main Chain	Fee / KB	0.00002023 tBTC
Relay time	Saturday, November 3rd 2018, 20:13:43 +01:00	Size	351 bytes
Time until confirmed	after 6 minutes		

1 INPUTS — Total Inputs: 0.25000000 tBTC

P2SH 2NEwEAG9VyFYt2sjLpuHrU4Abb7nGJfc7PR (0.25000000)

1 OUTPUTS

mqA1ojkoQqakFerGzsD7Kv7iFQjMVoFj4W (0.24999290)

INPUT SCRIPTS

OP_0
304402206ed14934f52527b8a34fe2a1fffbd53c98c88049e4ffe0f32c3196f6bdf02c4f022
06e5c2ae1d6c71a6262477303cf0fab82609c6c8f948fa7d3fef6c7809977f31401
304402205ac1485bf5a88f1771732925da62ed7cf56ba0f5d76e231604766cb0fb21addb
02203548c10e30b4f30c417b84579294913a95edf40c43da97308fc4df543accba0301
OP_0 OP_IF 2539df5b OP_CHECKLOCKTIMEVERIFY OP_DROP
022fd8c0455bede249ad3b9a9fb8159829e8cfb2c360863896e5309ea133d122f2
OP_CHECKSIG OP_ELSE OP_2
02b4e59070d367a364a31981a71fc5ab6c5034d0e279eecec19287f3c95db84aef
03cf15041579b5fb7accbac2997fb2f3e1001e9a522a19c83ceabe5ae51a596c7c OP_2
OP_CHECKMULTISIG OP_ENDIF

OUTPUT SCRIPTS

OP_DUP OP_HASH160 69bb74424f07d766128ce06970020067516fe959
OP_EQUALVERIFY OP_CHECKSIG

Block 03e889de

Height	23,882
Hash	03e889de4c2ae83efa45b53adf8fbb2890d2f16e1a42362eb9c0bc8a698b5d05
Parent	078cbb262b42657f219418bf1ed2f097c60fe3b14c97b66deac52692fd6984b7
Child(ren)	0263351821e95c1ad9c93136c78c3e034f458d4e778d1908e00ed852489555fc
Version	1
Time	2018-11-03 18:40:52 UTC
Mined in	1m 1s
PoW algorithm	Secondary
Secondary scale	3,324
Edge bits	29
Nonce	13145007527189516869
Solution difficulty	217,724
Network difficulty	102,126
Total difficulty	1,246,983,019
Kernel offset	bf361013a24247fff5001ed401d197714c9bb212a9d3f87878d33567e6ae2b12
Total kernel offset	1b745cdd667c5184c32dfb7615293e9148745fa8009c60e3694b0f48606229e7
Reward	60 + 0.004 Grin

Kernels (2)

#	Type	Excess	Fee	Lock height
0	Coinbase	08464989659d1223834139eca9f8db8a3e1011f2e1841bb1e55364ee8fa2d4d90f		23,882
1		094f7e47add827f973273fa8cccd31751d41c0905812df09e5ff84d1411d9e92a5	0.004	23,862

Inputs (1)

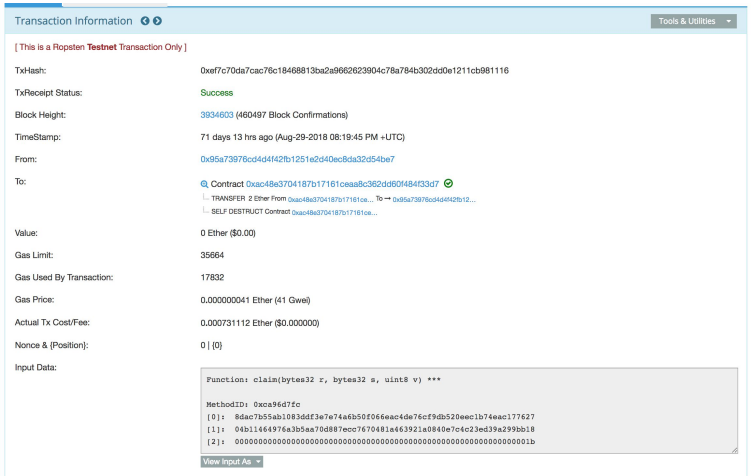
#	Commit
0	085a021dffae47e92105b2ce6d317fa5755cd77759106bb9cac86e99ada9269c2c

Outputs (2)

#	Type	Commit
0		083cab62ebc6f304324d61f59d35ffb956dd000d1e1f7132bc9217994b49179cc3
1	Coinbase	08d8d7b528df0541031963180cedd920dc984b8979cbfaf9dbf11ed1abfe9be2fb

Atomic swaps on testnet

T3: first Grin-tETH swap



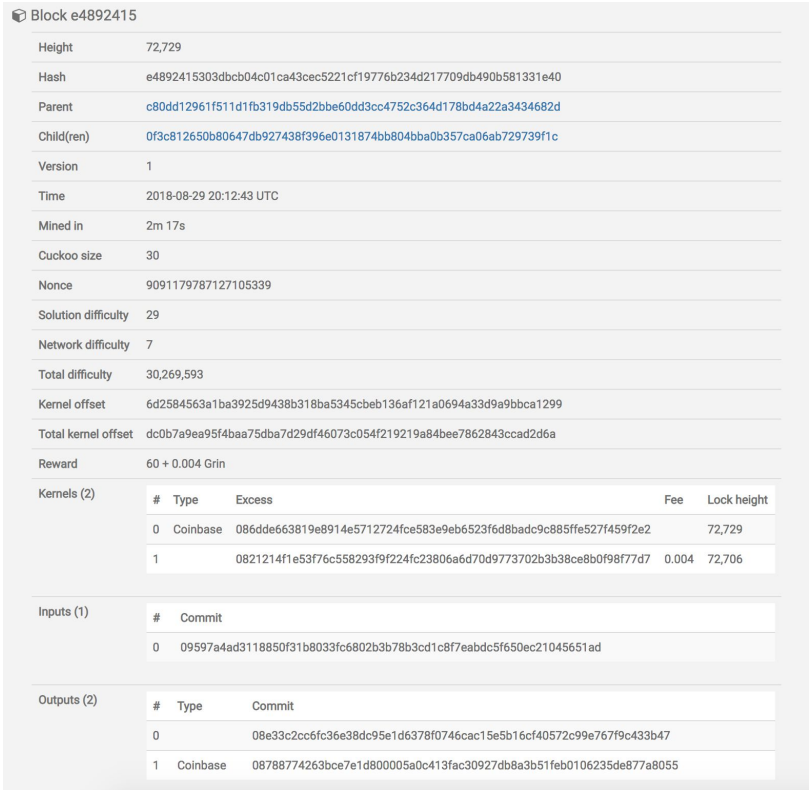
Transaction Information

Tools & Utilities

[This is a Ropsten **Testnet** Transaction Only]

TxHash:	0xef7c70da7cac76c18468813ba2a9662623904c78a784b302dd0e1211cb981116
TxReceipt Status:	Success
Block Height:	3934603 (460497 Block Confirmations)
TimeStamp:	71 days 13 hrs ago (Aug-29-2018 08:19:45 PM +UTC)
From:	0x95a73976cd4d4f42fb1251e2d40ec8da32d54be7
To:	Contract 0xac48e3704187b17161ceaa8c362dd60f484f33d7 TRANSFER 2 Ether From 0xac48e3704187b17161ce... To 0x95a73976cd4d4f42fb12... SELF DESTRUCT Contract 0xac48e3704187b17161ce...
Value:	0 Ether ($0.00)
Gas Limit:	35664
Gas Used By Transaction:	17832
Gas Price:	0.000000041 Ether (41 Gwei)
Actual Tx Cost/Fee:	0.000731112 Ether ($0.000000)
Nonce & {Position}:	0 | {0}
Input Data:	Function: claim(bytes32 r, bytes32 s, uint8 v) *** MethodID: 0xca96d7fc [0]: 8dac7b55ab1083ddf3e7e74a6b50f066eac4de76cf9db520eec1b74eac177627 [1]: 04b1464976a3b5aa70d887ecc7670481a463921a0840e7c4c23ed39e299bb18 [2]: 000000000000000000000000000000000000000000000000000000000000001b

View Input As



Block e4892415

Height	72,729
Hash	e4892415303dbcb04c01ca43cec5221cf19776b234d217709db490b581331e40
Parent	c80dd12961f511d1fb319db55d2bbe60dd3cc4752c364d178bd4a22a3434682d
Child(ren)	0f3c812650b80647db927438f396e0131874bb804bba0b357ca06ab729739f1c
Version	1
Time	2018-08-29 20:12:43 UTC
Mined in	2m 17s
Cuckoo size	30
Nonce	9091179787127105339
Solution difficulty	29
Network difficulty	7
Total difficulty	30,269,593
Kernel offset	6d2584563a1ba3925d9438b318ba5345cbeb136af121a0694a33d9a9bbca1299
Total kernel offset	dc0b7a9ea95f4baa75dba7d29df46073c054f219219a84bee7862843ccad2d6a
Reward	60 + 0.004 Grin

Kernels (2)

#	Type	Excess	Fee	Lock height
0	Coinbase	086dde663819e8914e5712724fce583e9eb6523f6d8badc9c885ffe527f459f2e2		72,729
1		0821214f1e53f76c558293f9f224fc23806a6d70d9773702b3b38ce8b0f98f77d7	0.004	72,706

Inputs (1)

#	Commit
0	09597a4ad3118850f31b8033fc6802b3b78b3cd1c8f7eabdc5f650ec21045651ad

Outputs (2)

#	Type	Commit
0		08e33c2cc6fc36e38dc95e1d6378f0746cac15e5b16cf40572c99e767f9c433b47
1	Coinbase	08788774263bce7e1d800005a0c413fac30927db8a3b51feb0106235de877a8055

Code

Proof-of-concept code is available on Github

Can be used to perform Grin-BTC/ETH swaps on T4

4 round trips of communication

```
(ENV) grinswap:~/MW/grinswap$ ./swap sell
################################
# Grin -> BTC/ETH atomic swap
#
# This script is used for selling grin coins for Bitcoin or Ether through an atomic swap
#
# What is the name of the wallet you want to use? wallet
# How much grin do you want to sell? 100
# Which currency would you like to receive? [BTC/ETH]: BTC
# How much BTC do you want to receive? 0.1
# At which Bitcoin address would you like to receive this? mmzi4QaUwXfA3vncequpQbDiasGUn2GA57
# What is the height of the last Grin T4 block? 30509
#
# Created file 'ac28de17_seller_1.json', please send it to the buyer
################################
```

That’s all!

Reading material

Adaptor signatures / scriptless scripts:

https://www.youtube.com/watch?v=ovCBT1gyk9c
https://joinmarket.me/blog/blog/flipping-the-scriptless-script-on-schnorr
https://lists.launchpad.net/mimblewimble/msg00086.html

Grin-ETH atomic swap on T3:

https://medium.com/grinswap/first-grin-atomic-swap-a16b4cc19196

Proof-of-concept code:

https://github.com/GrinSwap/proof-of-concept

Q&A

https://grin-tech.org