Revokable branched outputs (Draft 1)

Abstract / Introduction

While to this day Monero remains one of the top privacy coins in the cryptocurrency space it still lacks in several aspects, one important aspect that is drawing a lot of investment and development to other projects is smart contracts and scripting. While scripting as implemented in Ethereum and Bitcoin could not easily be implemented in Monero without heavily compromising the anonymity guarantees that Monero gives its users, increased interoperability or customizability of the behavior of outputs should still be sought after. This proposal shows how a relatively simple extension to Monero transactions using already used cryptographic primitives can lead to more variable applications of Monero while still preserving privacy and decentralization. This feature would enable atomic swaps and more complex escrow/multisig behavior.

1 Proposal / mechanism

The proposal would consist of the following modifications:

1. Every transaction output requires a further sG revocation key
2. Spending an output no longer only requires the knowledge of the one time address private key k^o but also the secret s used to create the secondary public revocation key sG (it's realized that also the mask and commitment is required to spend outputs in Monero but for the moment just the key side of things are going to be considered)
3. s also needs to be used to sign the ring signature and its key image $\tilde{K}_s$ is also included as part of the input
4. A reused key image $\tilde{K}_s$ is invalid
5. The verification of commitments is changed: instead of summing all output commitments, input commitments and miner fee the sum of all outputs may be greater than inputs **only if** the duplicate sums have the same public revocation key sG. Like this as soon as one duplicate output is spent the other is automatically made invalid because they would share the same key image $\tilde{K}_s$
6. Transactions may not reuse any previously existing sG

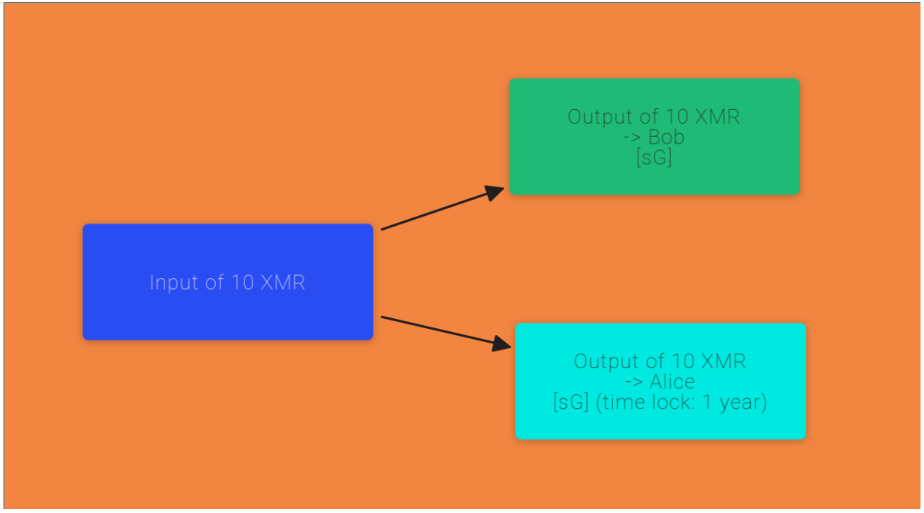
2 Anonymity

Since all transactions would have to have this additional revocation key and it would be part of the ring signature the general level of anonymity of transactions should remain the same. Furthermore this mechanism would allow the creation of decoy outputs possibly increasing anonymity. The ability to create more versatile transaction would likely lead to new heuristics but that is to be expected.

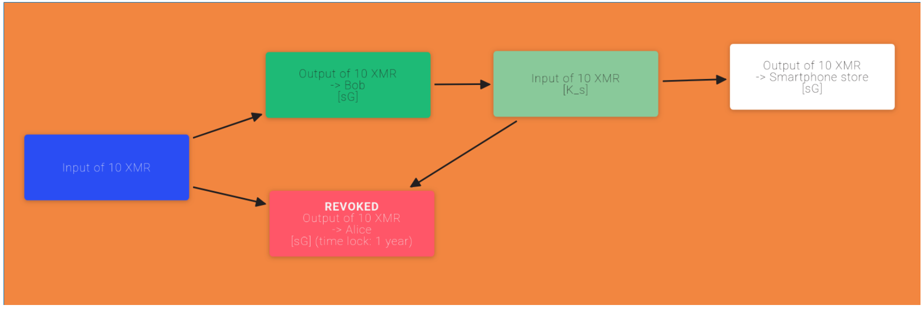
3 Utility

3.1 Timelocked inheritance

Let's say Bob owns some XMR but he doesn't trust his daughter so he can't just give her his keys because she might take his funds while he's alive. Nevertheless he wants his daughter to have his funds should anything happen to him. Let's say Bob has a child Alice to which he wants his 10 XMR to go to if he were to be gone for more than a year, so he constructs a transaction as follows:



Now let's imagine Bob wants to use his money to pay for a smartphone:



Now because the key image of the revocation key was used Alice can no longer spend the output even after a year has passed because doing so would require her to use the revocation key which was already used so the network would simply reject her transaction. If Bob didn't spend his Monero for a year because say he unfortunately got hit by a bus Alice could now spend the output she inherited, revoking the other transaction. (Note: the sG in the transaction to the smartphone store would be different).

3.2 Atomic swaps

The transaction construction that was just presented could also be used for atomic swaps, one would just need a smart contract capable of verifying the public revocation key. Bitcoin's scripting language doesn't yet have any opcodes that allow it to do some calculations based on elliptic curve points. So while directly swapping with Bitcoin might not be possible at the beginning one could likely use an Ethereum smart contract as an intermediary. For simplicities sake we'll imagine that Bitcoin implements a new opcode `OP_CHECK_ED25519_POINT` that takes a private and public key and verifies that they belong to each other. Like in the previous scenario let's imagine that Alice has Monero and wants to trade with Bob who has Bitcoin. After they've agreed on a price Alice can use some Monero to create 2 outputs that revoke each other based on a randomly generated secret s. One output will be a timelocked refund transaction to herself, the other an output to Bob. After she has broadcasted this transaction she can send Bob sG who can create an output locked by the following locking script:

```
<sG> OP_CHECK_ED25519_POINT OP_IF
  OP_TRUE
OP_ELSE
  <lock time> OP_CHECKLOCKTIMEVERIFY OP_DROP
  //normal P2PKH script
  OP_DUP OP_HASH160 <Bob's pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG
OP_ENDIF
```

Since all revocable outputs act like 2-of-2 multisig outputs Bob doesn't have to worry that someone else will be able to redeem his Monero once Alice publicizes s on the Bitcoin blockchain.

3.3 Escrow

Just as in the inheritance scenario one can combine this new capability with multisig wallets to create even more versatile escrow transactions.

4 Advantages

The addition of revocation keys would allow for more versatile and complex transactions while preserving privacy. Furthermore all the outputs of these more complex transactions can be used as decoys in other transactions assuming that their timelocks have expired at the time they are selected as decoys.

5 Disadvantages

The addition of another key to the ring signatures would lead to a size increase of outputs increasing fees and increasing the size of the blockchain.