Manta Workflow

www.manta.network **version 1.0**, March 6, 2021

1 How does Manta work?

We use a toy example to demonstrate a typical Manta workflow. For details of Manta protocol, see [CXZ20]. Figure 1 shows the workflow of a user using Manta for private payment and exchange. For simplicity, we assume there are two sets of base coins that Manta supports: $CoinA$ and $CoinB$, each of which is a public coin in Manta connected public ledgers (e.g. Polkadot relaychain or parachain [dot]). Extension to multiple base coins is straightforward. In this example, Alice starts with a wallet, which holds 20 $CoinA$s under a public address $\texttt{PAddr}_1$, and a private coin $pCoinA_0$, with a face value of 30 $CoinA$s.

Mint. Now, Alice wants to mint her public coin $CoinA_1$, under a public address $\texttt{PAddr}_1$, to a private coin $pCoinA_1$. Alice generates a private coin $pCoinA_1$ in her wallet. $pCoinA_1$ consists of the following parts:

- $\texttt{SAddr}_1$, a secretive address that is never revealed. This is similar to "Shielded Address" in ZCash [BCG+14].
- cm_{A_1}, a commitment to $pCoinA_1$'s amount (in this case 20), its void number (vn_{A_1}), and some auxiliary data.
- vn_{A_1}, a void number that is unique to $pCoinA_1$ and is only revealed when the $pCoinA_1$ is spent (either transferred, forfeited to public coin, or exchanged).
- 20, the nomination of this private coin.

Then, Alice sends a $\mathsf{tx}_{\mathsf{mint}}$ that contains the public address ($\texttt{PAddr}_{A_1}$, may also be referred to as *public key* in other contexts), the commitment to the private coin (cm_{A_1}), and the mint amount (10). Upon receiving $\mathsf{tx}_{\mathsf{mint}}$, the ledger will update its state by including the commitment cm_{A_1} to its $\texttt{CMList}$, and increase the pool size by 20.

We remark that it is impossible to link $pCoinA_0$ and $pCoinA_1$ in the above example.

Transfer. Next, suppose Alice wishes to conduct a private transfer with her private coins, for example, $pCoinA_0$ and $pCoinA_1$. Alice pours these private coins to new private addresses. For simplicity, in this example, we assume that these new secretive addresses are till owned by herself. A transfer transaction takes two private coins and will produce two new private coins with new addresses. More specifically, she sends $\mathsf{tx}_{\mathsf{transfer}}$ to Manta ledger that contains:

- cm_{A_2} and cm_{A_3}, the commitments to the new coins.
- vn_{A_0} and vn_{A_1}, the void numbers of the old coins. Now, these void numbers are revealed and voided.
- $\pi_{\{A_0,A_1\}\rightarrow\{A_2,A_3\}}$, a zero-knowledge proof that proves the transaction is valid and authenticated (more details will be given later).

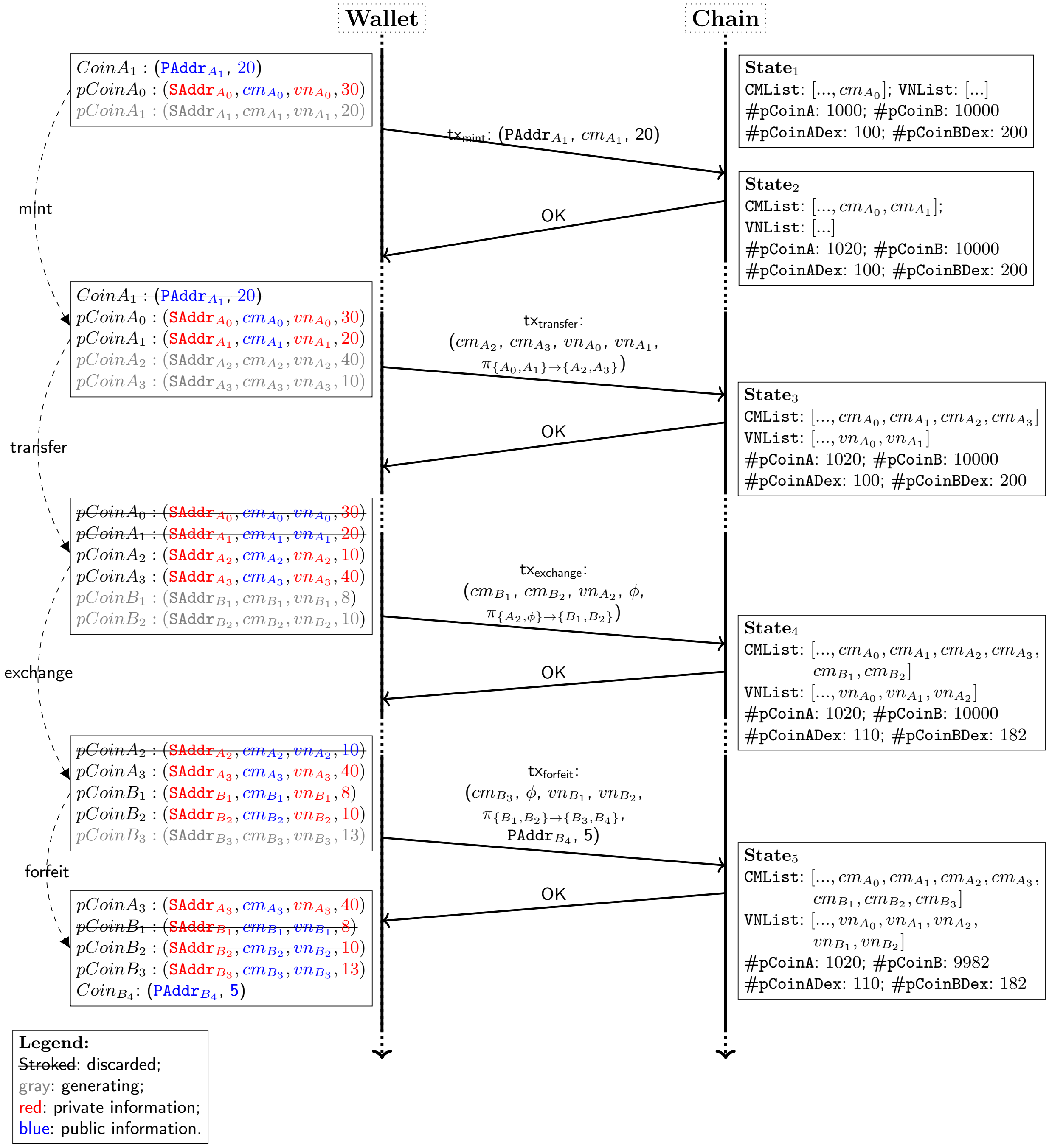
Upon the recipient of the transaction, Manta ledger checks validity of the zero-knowledge proof ($\pi_{\{A_0,A_1\}\rightarrow\{A_2,A_3\}}$) and ensures that vn_{A_0} and vn_{A_1} have never been revealed before. Note, since we have revealed only vn_{A_0} and vn_{A_1} (instead of cm_{A_0} and cm_{A_1}), and a ZKP ($\pi_{\{A_0,A_1\}\rightarrow\{A_2,A_3\}}$), the adversary cannot link the revealed void numbers (vn_{A_0} and vn_{A_1}) with the commitments of the old coins (cm_{A_0} and cm_{A_1}).

Exchange. Exchange works similarly as transfer. The main difference is that, in an exchange, a new type of coin $pCoinB$ is output, and the ZKP includes a statement that this exchange preserves the Automatic Market Maker's (AMM, see [But18]) ledger invariant, i.e., $\#\texttt{pCoinADex} \times \#\texttt{pCoinBDex} = k$.

Concretely, in our case, $k = 2 \times 10^4$. Alice wants to exchange 10 $CoinA$ to $CoinB$. To maintain a constant invariant, Alice is expecting 18 $CoinB$s in return (rounding to integers for simplicity), for which she will split into two coins, denoted by $pCoinB_1$ and $pCoinB_2$.

Note that the result from [AEC21, Whi20] claimed that privacy preserving AMM is impossible. Therefore we can assume that the attacker learns that the value of $pCoinA_2$ is 10; that of $pCoinB_1$ and $pCoinB_2$

Fig. 1: Manta overview


Wallet
Chain
$CoinA_1$: ($\texttt{PAddr}_{A_1}$, 20)
$pCoinA_0$: ($\texttt{SAddr}_{A_0}$, cm_{A_0}, vn_{A_0}, 30)
$pCoinA_1$: ($\texttt{SAddr}_{A_1}$, cm_{A_1}, vn_{A_1}, 20)
mint
~~$CoinA_1$: ($\texttt{PAddr}_{A_1}$, 20)~~
$pCoinA_0$: ($\texttt{SAddr}_{A_0}$, cm_{A_0}, vn_{A_0}, 30)
$pCoinA_1$: ($\texttt{SAddr}_{A_1}$, cm_{A_1}, vn_{A_1}, 20)
$pCoinA_2$: ($\texttt{SAddr}_{A_2}$, cm_{A_2}, vn_{A_2}, 40)
$pCoinA_3$: ($\texttt{SAddr}_{A_3}$, cm_{A_3}, vn_{A_3}, 10)
transfer
~~$pCoinA_0$: ($\texttt{SAddr}_{A_0}$, cm_{A_0}, vn_{A_0}, 30)~~
~~$pCoinA_1$: ($\texttt{SAddr}_{A_1}$, cm_{A_1}, vn_{A_1}, 20)~~
$pCoinA_2$: ($\texttt{SAddr}_{A_2}$, cm_{A_2}, vn_{A_2}, 10)
$pCoinA_3$: ($\texttt{SAddr}_{A_3}$, cm_{A_3}, vn_{A_3}, 40)
$pCoinB_1$: ($\texttt{SAddr}_{B_1}$, cm_{B_1}, vn_{B_1}, 8)
$pCoinB_2$: ($\texttt{SAddr}_{B_2}$, cm_{B_2}, vn_{B_2}, 10)
exchange
~~$pCoinA_2$: ($\texttt{SAddr}_{A_2}$, cm_{A_2}, vn_{A_2}, 10)~~
$pCoinA_3$: ($\texttt{SAddr}_{A_3}$, cm_{A_3}, vn_{A_3}, 40)
$pCoinB_1$: ($\texttt{SAddr}_{B_1}$, cm_{B_1}, vn_{B_1}, 8)
$pCoinB_2$: ($\texttt{SAddr}_{B_2}$, cm_{B_2}, vn_{B_2}, 10)
$pCoinB_3$: ($\texttt{SAddr}_{B_3}$, cm_{B_3}, vn_{B_3}, 13)
forfeit
$pCoinA_3$: ($\texttt{SAddr}_{A_3}$, cm_{A_3}, vn_{A_3}, 40)
~~$pCoinB_1$: ($\texttt{SAddr}_{B_1}$, cm_{B_1}, vn_{B_1}, 8)~~
~~$pCoinB_2$: ($\texttt{SAddr}_{B_2}$, cm_{B_2}, vn_{B_2}, 10)~~
$pCoinB_3$: ($\texttt{SAddr}_{B_3}$, cm_{B_3}, vn_{B_3}, 13)
$Coin_{B_4}$: ($\texttt{PAddr}_{B_4}$, 5)
$\mathsf{tx}_{\mathsf{mint}}$: ($\texttt{PAddr}_{A_1}$, cm_{A_1}, 20)
OK
$\mathsf{tx}_{\mathsf{transfer}}$: ($cm_{A_2}$, cm_{A_3}, vn_{A_0}, vn_{A_1}, $\pi_{\{A_0,A_1\}\to\{A_2,A_3\}}$)
OK
$\mathsf{tx}_{\mathsf{exchange}}$: ($cm_{B_1}$, cm_{B_2}, vn_{A_2}, ϕ, $\pi_{\{A_2,\phi\}\to\{B_1,B_2\}}$)
OK
$\mathsf{tx}_{\mathsf{forfeit}}$: ($cm_{B_3}$, ϕ, vn_{B_1}, vn_{B_2}, $\pi_{\{B_1,B_2\}\to\{B_3,B_4\}}$, $\texttt{PAddr}_{B_4}$, 5)
OK
State$_1$
CMList: [..., cm_{A_0}]; VNList: [...]
#pCoinA: 1000; #pCoinB: 10000
#pCoinADex: 100; #pCoinBDex: 200
State$_2$
CMList: [..., cm_{A_0}, cm_{A_1}];
VNList: [...]
#pCoinA: 1020; #pCoinB: 10000
#pCoinADex: 100; #pCoinBDex: 200
State$_3$
CMList: [..., cm_{A_0}, cm_{A_1}, cm_{A_2}, cm_{A_3}]
VNList: [..., vn_{A_0}, vn_{A_1}]
#pCoinA: 1020; #pCoinB: 10000
#pCoinADex: 100; #pCoinBDex: 200
State$_4$
CMList: [..., cm_{A_0}, cm_{A_1}, cm_{A_2}, cm_{A_3}, cm_{B_1}, cm_{B_2}]
VNList: [..., vn_{A_0}, vn_{A_1}, vn_{A_2}]
#pCoinA: 1020; #pCoinB: 10000
#pCoinADex: 110; #pCoinBDex: 182
State$_5$
CMList: [..., cm_{A_0}, cm_{A_1}, cm_{A_2}, cm_{A_3}, cm_{B_1}, cm_{B_2}, cm_{B_3}]
VNList: [..., vn_{A_0}, vn_{A_1}, vn_{A_2}, vn_{B_1}, vn_{B_2}]
#pCoinA: 1020; #pCoinB: 9982
#pCoinADex: 110; #pCoinBDex: 182
Legend:
~~Stroked~~: discarded;
gray: generating;
red: private information;
blue: public information.

combined is 18. Nonetheless, $pCoinA_2$ remains anonymous, since only vn_{A_2} is revealed. On the other hand, both face values and the addresses of $pCoinB_1$ and $pCoinB_2$ remain private.

Forfeit. Forfeit is a process through which the user claims back the base coin from the private coin. In our example, Alice wants to forfeit 5 $CoinB$s, and put the reminder of $pCoinB_1$ and $pCoinB_2$ to $pCoinB_3$. The workflow is identical to a transfer workflow, with the only different that one of the output coins, i.e., $Coin_{B_4}$, is public,.

2 Manta's security guarantee

Now, we briefly explain why Manta's payment and exchange is private. We can make the following observations from Figure 1:

1. The secretive addresses (this is similar to shielded address in ZCash), e.g. $\texttt{SAddr}_i$ are never revealed.
2. The public information, e.g., *cm*-s and vn-s cannot be linked.
3. All operations (except for mint) consume two old coins (UTXOs) and generate two new coins (UTXOs). Under the assumption that mint leaks the value of a single commitment, and exchange leaks the values of the summation of the two commitment (exchange have to leak the value of the trade otherwise the participant would be blind on the exchange ratio), the overall scheme remains still private. Even more so, when conducting a transfer, exchange, or forfeit, user does not identify which commitment is been used; instead, she proves that " I own a certain commitment that was submitted to the ledger earlier, whose the void number is vn". Thus, knowing the face value of a commitment does not help the attacker de-anonymize the sender.

This also explains why Manta's privacy guarantee is not contradict with the negative theoretical results of private AMM [AEC21]. In a nutshell, Manta guards the privacy despite the leak of trading price.

References

AEC21. Guillermo Angeris, Alex Evans, and Tarun Chitra. A note on privacy in constant function market makers, 2021.

BCG+14. Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *IEEE Symposium on Security and Privacy*, pages 459–474. IEEE Computer Society, 2014.

But18. Vitalik Buterin. Improving front running resistance of x*y=k market makers. https://ethresear.ch/t/improving-front-running-resistance-of-x-y-k-market-makers/1281, 2018.

CXZ20. Shumo Chu, Qiudong Xia, and Zhenfei Zhang. Manta: Privacy preserving decentralized exchange. Cryptology ePrint Archive, Report 2020/1607, 2020. https://eprint.iacr.org/2020/1607.

dot. Polkadot: Decentralized web 3.0 blockchain interoperability platform. https://polkadot.network/.

Whi20. Barry WhiteHat. Why you can't build a private uniswap with zkps., 2020.