Wanchain cross-chain procedure

CONTENTS

1.HTLC procedure

A Hashed Timelock Contract (HTLC) is a conditional transfer that use hashlock and timelock to require that the receiver of a payment either acknowledge receiving the payment prior to a deadline by generating cryptographic proof of payment or forfeit the ability to claim the payment, returning it to the payer.

When the transfer is prepared, the payer's funds are put on hold by the contract, pending the fulfillment of a predefined condition. The condition is a hashlock, and the contract stipulates that the recipient may claim the funds by presenting a valid preimage of the hashlock before a given timeout. After the timeout, the payer can get the funds back.

2.Cross Transaction Process

HTLC is used in Wanchain cross chain system, two HTLC dividedly deployed on cross-chain and Wanchain will ensure the secure cross-chain transfer.

2.1 Inbound transmission process

Here we take the user Alice, who wants to transfer value from an original chain to Wanchain, as an example to illustrate the asset inbound transmission process:

- ◆ step1: Lock Funds

Alice initiates a transaction OriTx with her OriAccount on the original chain to send an asset to HTLC on Originalchain and broadcasts the cross-chain transaction. The OriTx would contain one cryptographic proof hashX. The asset would be locked in HTLC Account and registered with hashX and added in the registry.

The transaction includes the detail data, such as:

OriTx: OriAccount, WanAccount, StoremanGroupAccont, value, hashX

When the transaction been scanned by StormanGroup, after the transaction has been confirmed on the original chain, a transaction WanTx would be sent to HTLC on Wanchain to lock it's certain quota. The WanTx would also contain hashX, and the quota will be registered with hashX and added in the registry.

The transaction includes the detail data, such as:

```
WanTx: WanAccount, value, hashX
```

- ◆ step2: Redeem Funds

After the WanTx had been confirmed on the Wanchain, Alice can send a transaction WanTx2 with her WanAccount on Wanchain to redeem the cross-chain asset. In this transaction, Alice need to provide X, the valid preimage of hashX, in WanTx2 to claim the token value redeem on Wanchain before the timeout. The HTLC on Wanchain will get the hash value of X, and then check the quota registry of the original chain by hashX, if exist, the contract will confirm the validity of WanTx2. If the transfer had been validated, the contract will mint token to WanAccount. If WanTx2 becomes invalid, Alice need re-initiate the transaction.

The transaction includes the detail data, such as:

```
WanTx2: X
```

Storeman Group will scan the redeem transaction of Alice, and get X after the transaction been confirmed. Then Storeman Group will initiate a redeem transaction OriTx2 to claim the asset on HTLC on Originalchain before the timeout. The contract gets the hash value of X, and checks the asset registry by hashX, if exist, the locked asset will be transferred to StoremanGroup. If OriTx2 become invalid, the StoremanGroup will re-initiate the transaction.

The transaction includes the detail data, such as:

```
OriTx2: X
```

- ◆ step3: Revoke Funds

After the timeout, if Alice still not take the redeem action, StoremanGroup will initiate a revoke transaction WanTx3 to take it's locked quota back from HTLC on Wanchain. The contract will check the registry status of hashX, if not been redeemed; the contract will return back the quota to StoremanGroup.

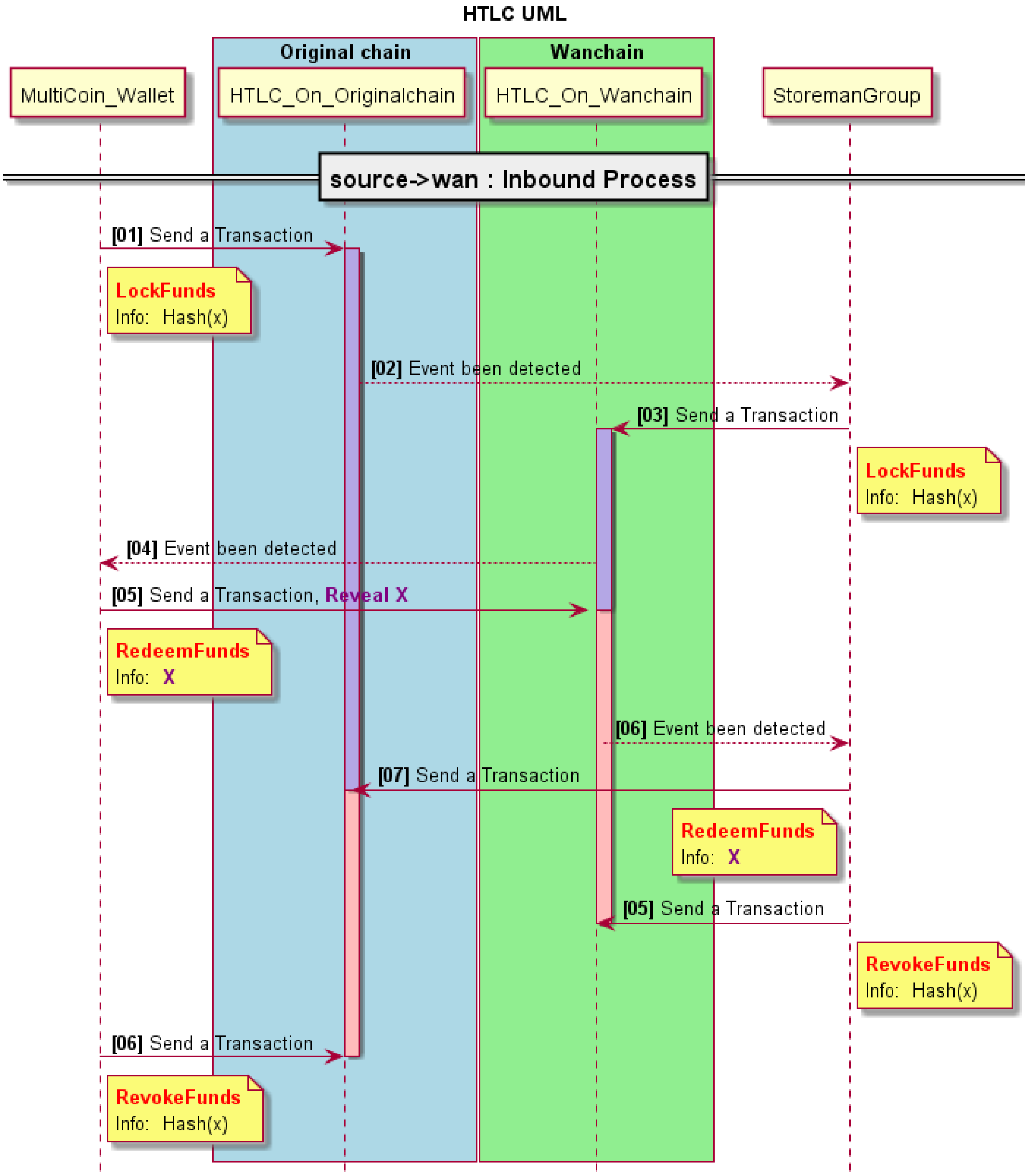
The transaction includes the detail data, such as:

WanTx3: hashX

If Alice didn't redeem the token value on Wanchain, she needs to initiate a revoke transaction OriTx3 to get her locked asset back from HTLC on Originalchain. HashX should be included in the transaction, and the contract check the status of the registry for hashX, if valid, the asset will be transfer back to her OriAccount.

The transaction includes the detail data, such as:

OriTx3: hashX



2.2 Outbound transmission process

In the next example, we have user Bob transferring Value from Wanchain back to the original chain to illustrate the asset outbound transmission process:

- step1: Lock Funds

Bob initiates a transaction WanTx with his WanAccount on the Wanchain to send the certain token value to HTLC on Wanchain and broadcasts the cross-chain transaction. The WanTx would contain one cryptographic proof hashX. The token value would be locked in HTLC Account and registered with hashX and added in the registry.

The transaction includes the detail data, such as:

WanTx: WanAccount, OriAccount, StoremanGroupAccont, value, hashX

When the transaction had been scanned by StormanGroup, after the transaction has been confirmed on the Wanchain, a transaction OriTx would be sent to HTLC on Originalchain to lock it's certain Original chain asset. The OriTx would also contain hashX, and the asset will be registered with hashX and added in the registry.

The transaction includes the detail data, such as:

OriTx: OriAccount, value, hashX

◆ step2: Redeem Funds

After the OriTx had been confirmed on the Original chain, Bob can send a transaction OriTx2 with his OriAccount on Original chain to redeem the asset. In this transaction, Bob need to provide X, the valid preimage of hashX, in OriTx2 to claim the asset redeem on Original chain before the timeout. The HTLC on Originalchain will get the hash value of X, and then check the asset registry of the Original chain by hashX, if exist, the contract will confirm the validity of OriTx2. If the transfer been validated, the contract will transfer asset back to OriAccount. Or if OriTx2 becomes invalid, Bob need re-initiate the transaction.

The transaction includes the detail data, such as:

OriTx2: X

StoremanGroup will scan the redeem transaction of Bob, and get X after the transaction been confirmed. Then StoremanGroup will initiate a redeem transaction WanTx2 to claim the token value on HTLC on Wanchain before the timeout. The contract gets the hash value of X, and checks the token registry by hashX, if exist, the locked quota will be transferred back to StoremanGroup, and the certain token will be burned. If OriTx2 become invalid, the StoremanGroup will re-initiate the transaction.

The transaction includes the detail data, such as:

WanTx2: X

- step3: Revoke Funds

After the timeout, if Bob still not take the redeem action, StoremanGroup will initiate a revoke transaction OriTx3 to take it's locked asset back from HTLC on Originalchain. The contract will check the registry status of hashX, if not been redeemed, the contract will return back the asset to StoremanGroup.

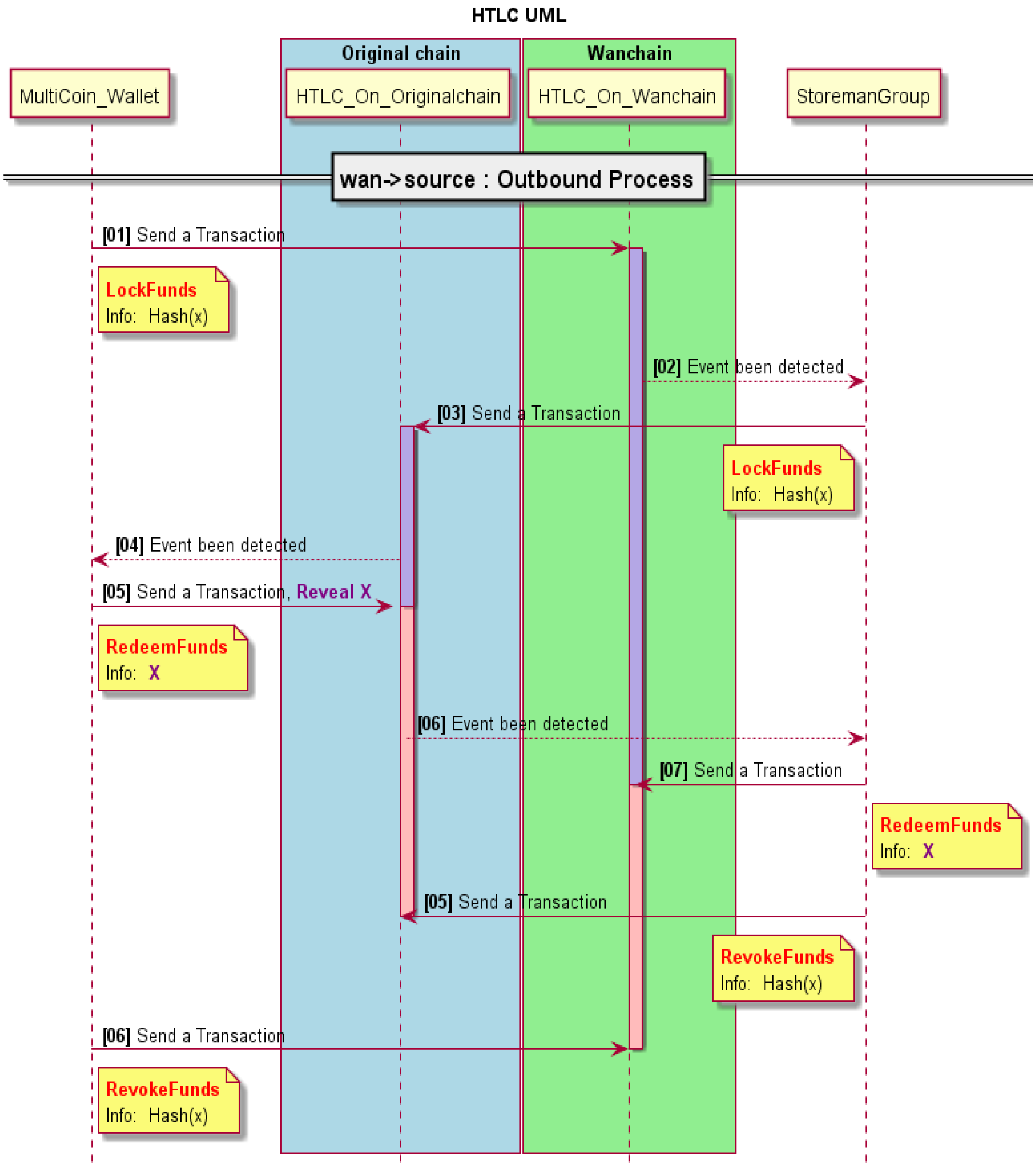
The transaction includes the detail data, such as:

OriTx3: hashX

If Bob didn't redeem the asset on Original chain, he needs to initiate a revoke transaction WanTx3 to get his locked token value back from HTLC on Wanchain. HashX should be included in the transaction, and the contract check the status of the registry for hashX, if valid, the token will be transfer back to his WanAccount.

The transaction includes the detail data, such as:

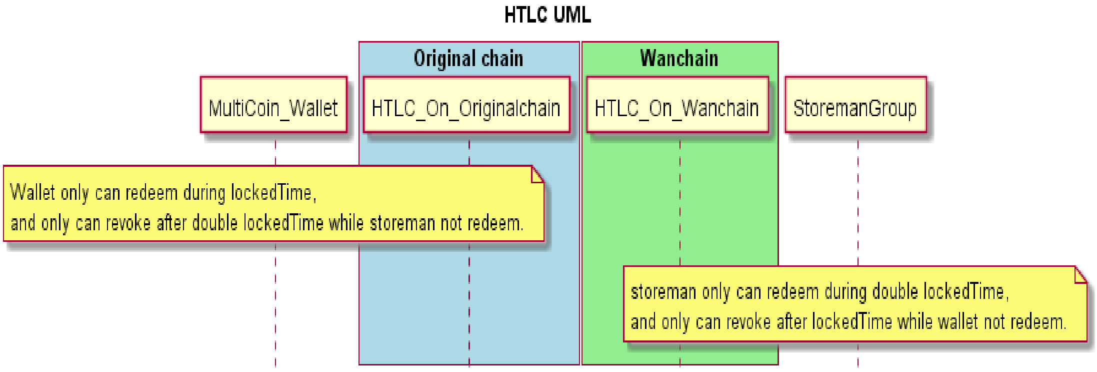
WanTx3: hashX



3.Time Slot

To protect the StoremanGroup, there is a litter difference of cross-chain process for user and StoremanGroup.

- User only can redeem during lockedTime, and only can revoke after double lockedTime while StoremanGroup not redeem.
- StoremanGroup only can redeem during double lockedTime, and only can revoke after lockedTime while user not redeem.

HTLC UML
Original chain
Wanchain
MultiCoin_Wallet
HTLC_On_Originalchain
HTLC_On_Wanchain
StoremanGroup
Wallet only can redeem during lockedTime,
and only can revoke after double lockedTime while storeman not redeem.
storeman only can redeem during double lockedTime,
and only can revoke after lockedTime while wallet not redeem.