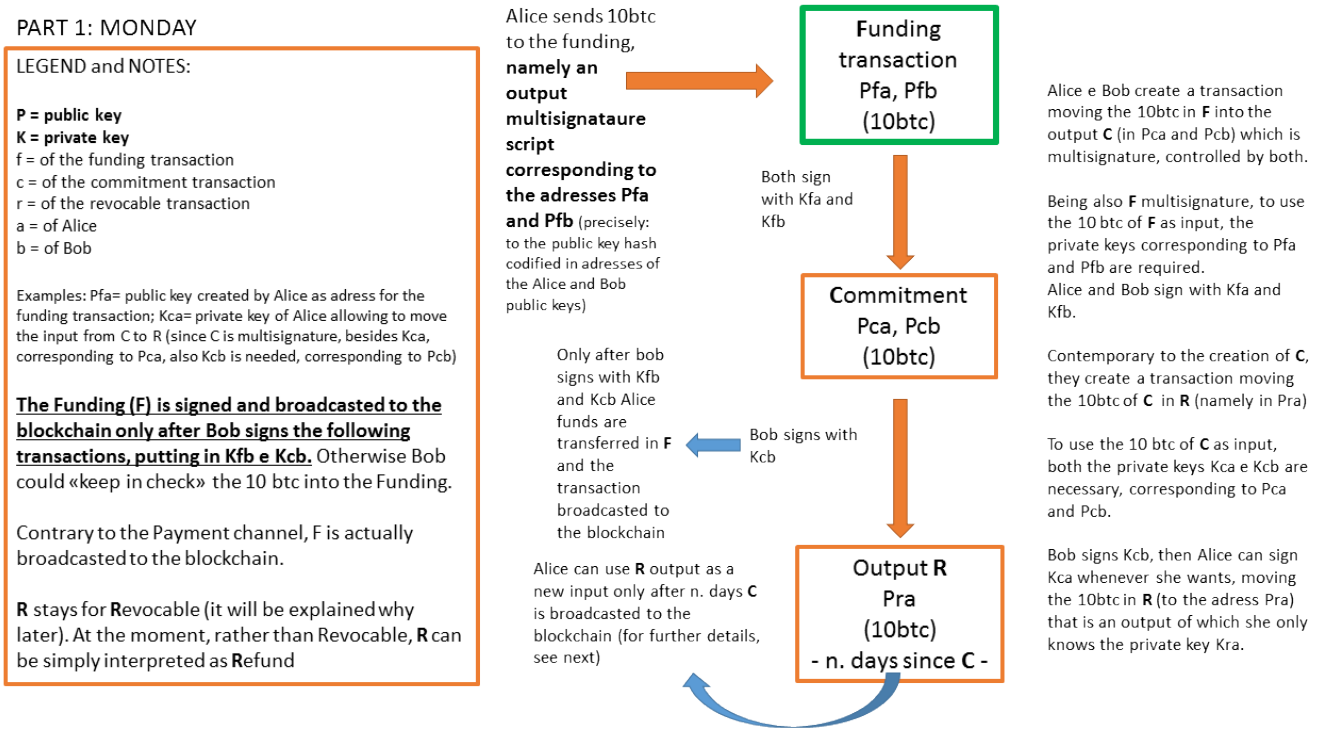
Lightning Network Part II

scritto da Alberto De Luigi | 26 Maggio 2016
< Back to Part I Go to Part III >

Permament relationships: the channel can remain open up to an indefinite time and does not require intermediaries



Lightning network requires a new protocol

F is the transaction «mother», **C** is **F**'s child and **R** is **C**'s child.

After **C** and **R** have been created and signed by Bob, **F** is broadcasted to the Blockchain. Alice doesn't broadcast before, otherwise she would lose her bitcoins – or more precisely – Bob could keep them «in check».

Since Bob already signed with his own private keys of **F** and **C**, Alice can sign and broadcast to the blockchain **C** and **R** whenever she wants.

As we see, a «child» can be created despite the input in the

«mother» transaction isn’t signed yet nor broadcasted to the blockchain. To do this it was necessary a fork allowing Segregated Witness (UPDATE: SegWit was approved in August 2017): with the new protocol a valid transaction can be broadcasted to the blockchain without the SIGNATURE SCRIPT: «SIGHASH NO_INPUT». This way the block can contain a transaction without its signing part.

Besides that, Lightning Network required other changes already implemented by means of a soft fork:

Soft fork of July 2016, block 419328 (BIP68,112,113)

Thanks to the soft fork of July 2016 at block #419328, besides nLocktime, another transaction parameter has been introduced: CheckLockSequenceVerify.

With this parameter it is possible to make available the output of a transaciton x after a certain number of blocks are created consequently to the registration into the blockchain of a given transaction y.

In the example above, the input of **R** can be used as output only after n. blocks (parameter «nSequence» subsequent to **C**.

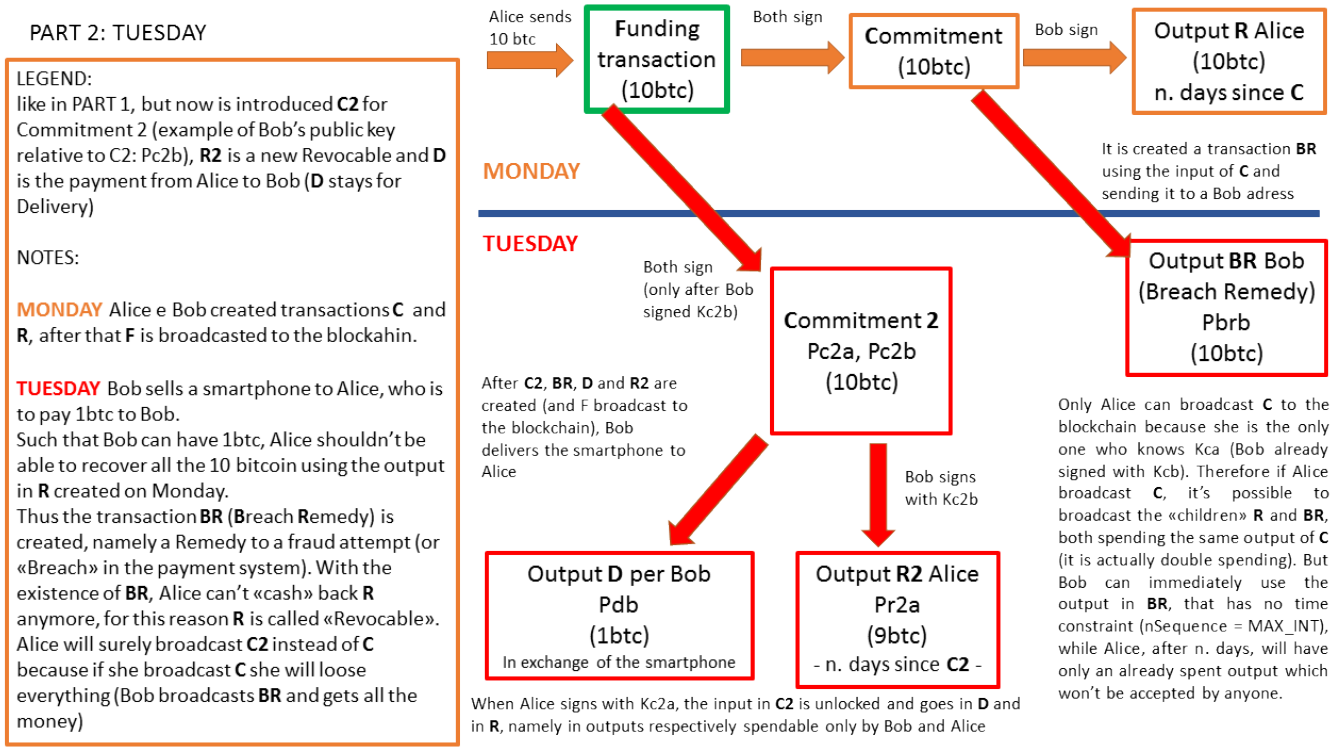
For example, if **C** is broadcasted on Monday, the 10btc in R can be used only since Wednsesday, if instead C is broadcasted on Wednesday, the 10btc in R can be used only Friday, and so on.

Until **C** is not broadcasted, the channel between Bob and Alice can remain open permanently.

As will be exposed, neither Bob nor Alice have interest in broadcasting **C** (except in cases of fraud/theft).

Lightning Network: how the channel works

That is, how Alice and Bob trade goods and services in exchange of bitcoins by means of a permanent bilateral channel opened with only one transaction **F** broadcasted to the blockchain



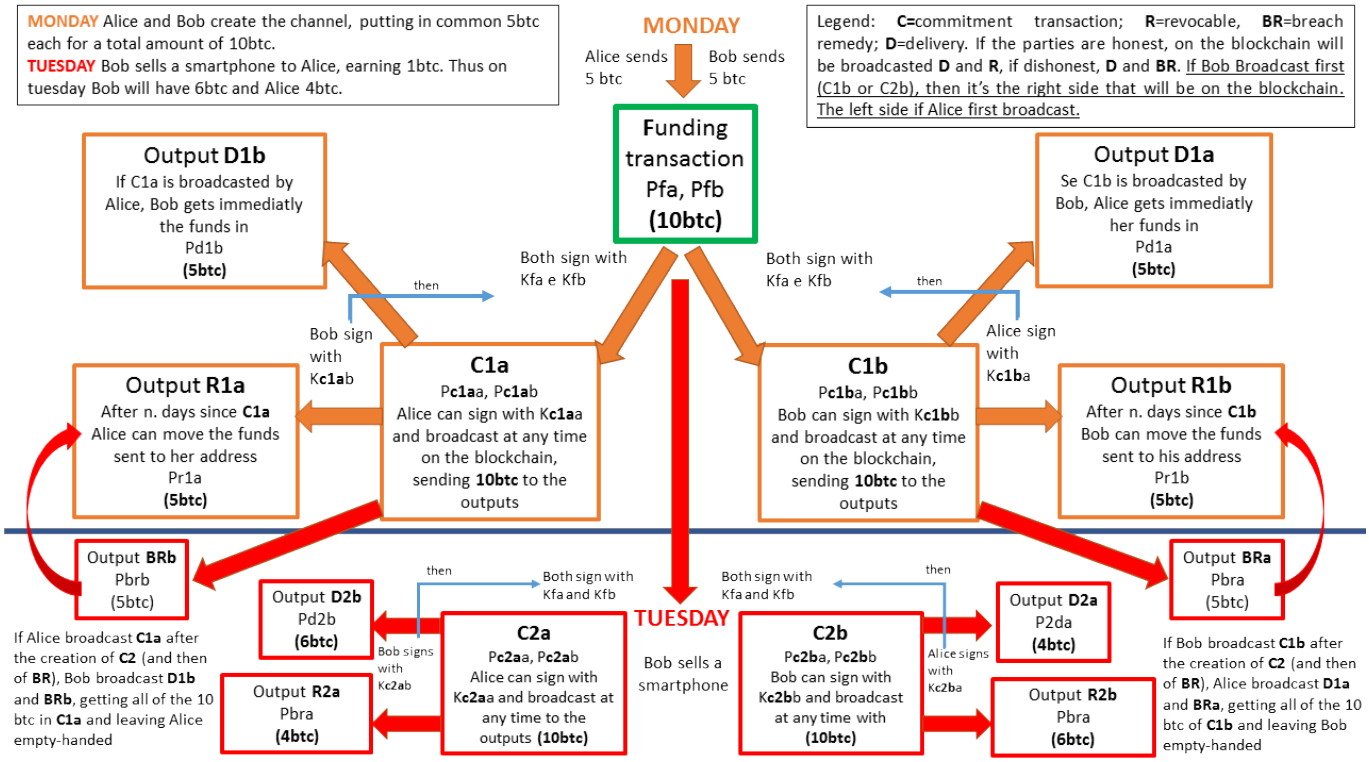
But it's not so simple…

In the scheme above, if Alice had to broadcast she would have economic convenience in transmitting **C2** instead of **C1**: broadcasting a previous status means to lose all the money because of the Breach Remedy. Despite this, **C2** could be never signed by Alice, if she disappears (maybe hit by a bus) or simply does not act in a rational way (she could even try to blackmail Bob keeping money in check…).

Moreover, Bob could have partecipated to the initial funding transaction: i.e. Bob and Alice could have put in common 5 bitcoin each to open the channel. In this case also Bob must have the guarantee to get back his money before to agree on the creation of the **F**unding transaction.

Therefore a «mirror scheme» is created, where each transaction has a pair. To understand the mechanics, see the next graph. To understand the notations, keep in mind these examples: since C1 is doubled into C1a and C1b, Alices' public key of C1a is «P**c1a**a», Bob's one «P**c1a**b», while Bob's private to move bitcoins from C1b is instead «K**c1b**b» and so on.

In the next graph Bob and Alice partecipate to the **F**unding with 5 bitcoin each.



A system of «sluice gates» of water channels

Look, for simplicity, only to the left side of the graph (the right side is specular, roles reversed):

– **before** Bob and Alice sign with Kfa and Kfb the transaction making the 10 btc **«flow»** from **F** to the address of **C1a**, Bob **already signed** with K**c1a**b the transaction sending the bitcoins from **C1a** to the outputs **D1b** and **R1a**

–At the same time, **before** Bob and Alice sign the transaction

F->**C2a** (with Kfa e Kfb), Bob **already signed** with K**c2a**b the transaction sending the bitcoins from **C2a** to the outputs **D2b** and **R2a**; moreover both already signed with Kc**1a**a and K**c1a**b the transaction sending the bitcoins from **C1a** to the output **BRb** (that is not time constrained, contrary to **R1a**)

Everything is possible thanks to the script «SIGHASH NO_INPUT» which requires a new Soft fork.

The term «to make flow» is not taken by chance, ***the system recalls the sluice gates of a water channel***: when Bob signs a transaction opens the downstream sluice gate, though the upstream gate is already closed. Once the upstream gate opens, the water (bitcoin) automatically streams in the open output downstream. In case two outputs are open to receive the «same water» (**R1a** e **BRb**) the one which has no time constraint nSequence (**BRb**) receives the water before, leaving the other output empty, though open.

Why to keep the channel open is convenient

After having bought the smartphone in exchange of 1btc, Alice would have convenience cheating Bob, broadcasting to the blockchain to a precedent state (the commitment C1 instead of C2) because in C1 Alice holds 5btc and Bob 5btc, while in C2 Alice 4btc and Bob 6btc. However, Alice besides broadcasting C2a can only broadcast C1a (she hasn't the private key Kfb to send funds in C1b), and if broadcast C1a Bob can get all 10btc: Bob in fact will be able to broadcast all the children of C1a,that is D1b e BRb, while Alice can only broadcast the child R1a. In this case BRb and R1a would represent a double spending, but BRb has not time constraint, thus it's spent far before R1a that is, precisely, «Revocable».

What happens if Alice brings back the smartphone because her sister doesn't like it? Simply, the new pair of transactions

C3a and C3b is created, whose outputs are 5btc to Alice and 5btc to Bob. Indeed, also the two new BR (Breach Remedy) will be created, not time contrained, which «invalidate» the Revocable txs children of C2, R2a and R2b, ensuring that no one broadcasts C2 instead of C3.

For both is convenient to keep the channel open, rather than broadcasting the transactions on the blockchain. In fact, who first broadcasts the last transaction **C**ommitment (suppose it is C2 in our graphical example) will have two disadvantages:

1) Pays the commission to the miner for the inclusion of the transaction in the blockchain;

2) Must wait the time constraint of the Revocable expires, before to spend the output for another transaction (while the other one can spend immediately, since the **D**elivery transaction is no time constrained).

In summary

Alice and Bob can exchange smartphone and bitcoin without never broadcast more than one transaction (the funding transaction) to the blockchain.

Between Alice and Bob a channel is open that could be permanent, until one of them try to cheat the other

Since this moment, each transaction between them (provided it is not of a greater amount than what deposited in the funding) will be off-chain.

This might reduce a lot (over time) the blockchain size, but everytime Alice or Bob make a new transaction with another person, like Charlie or Dave, they have to broadcast a new funding transaction. The blockchain size is still too large.

< Back to Part I Go to Part III >