# **Atomic Swap**

# **1 Basics**

A table with an overview of the support status and applicability.

|  |  |
| --- | --- |
| Status: | e.g. **Supported** |
| Architecture(s): |  |
| Component(s): | **Options include:** Python, Go, Javascript |
| Hardware: | *n/a* |

# **2 Overview**

Atomic swaps, or atomic cross-chain trading, is the exchange of one cryptocurrency to another cryptocurrency, without the need to trust a third-party. A relatively new piece of technology, atomic cross-chain trading is looking to revolutionize the way in which users transact with each other.

# **3 User details**

If Alice owned 5 Bitcoins but instead wanted 100 Litecoins, she would have to go through an exchange, i.e., a third-party. However, with atomic swaps, if Bob owned 100 Litecoins but instead wanted 5 Bitcoins, then Bob and Alice could make a trade. To prevent, for example, Alice accepting Bob’s 100 Litecoins but then failing to send over her 5 Bitcoins, atomic swaps utilizes what is known as hash time-locked contracts (HTLCs).

# **4 Technical details**

Hash time-locked contracts ensure that the atomic swap process is completely trustless by ensuring both fulfill the requirements of the trade. HTLCs require the recipient of payment to acknowledge receiving payment before a deadline by generating a cryptographic proof of payment. Or the recipient risks losing the right to the claim the payment, therefore returning the funds to the sender.

Therefore, for trade between Alice and Bob to take place, both must submit their transaction to their respective blockchain, Alice on the Bitcoin blockchain and Bob on the Litecoin blockchain. For Alice to claim the 100 Litecoins sent from Bob, she must produce a number that only she knows, used to generate a cryptographic hash, therefore providing proof of payment. Similarly, for Bob to claim the 5 Bitcoins that was sent from Alice, he must also provide the same number, that was used to generate the cryptographic hash.

# **5 Limitations**

There are some fundamental requirements for a cryptocurrency before it can successfully support atomic swaps. One such requirement is the implementation of the **Lightning network**.

If a hash time-locked contract can be thought of as linking two blockchains together, the lightning network can be thought of as linking payment channels together. That is, for Alice and Bob to transact with each other, they must be linked through payment channels. The lightning network allows for that.

Also, for a transaction to occur between two different blockchains, it is necessary for both blockchains to share the same cryptographic hash function, such as SHA-256. This is to allow for the hash-time locked contract to function properly when it comes to the user providing the number that was generated via the hash function.

# **6 Testing**

Testing this feature will likely be challenging and require access to a variety of test networks across the respective blockchains, including Divi’s.

# **7 Areas for improvement**

Additional blockchain conversions with a similar hash function could be a revolutionary improvement over the current methods available.

Feeless swaps would also be a very attractive improvement.

# **8 Known issues**

This technology is extremely new, and as such, new issues are still being documented. Most notably, network congestion across one chain or another (bitcoin) could create delays in payment, and in turn consumer doubtfulness.

# **9 References**

Decred Atomic Swaps - <https://github.com/decred/atomicswap>

Javascript (Ethereum) Atomic Swaps - <https://github.com/AltCoinExchange/ethatomicswap>

Python Atomic Swaps - <https://github.com/zack-bitcoin/ethereum-atomic-swap>

Atomic Swap information - <https://www.cryptocompare.com/coins/guides/what-are-atomic-swaps/>

Atomic Swap Network Congestion Issue - http://www.livebitcoinnews.com/barterdex-disables-bitcoin-based-atomic-swaps-due-network-congestion/