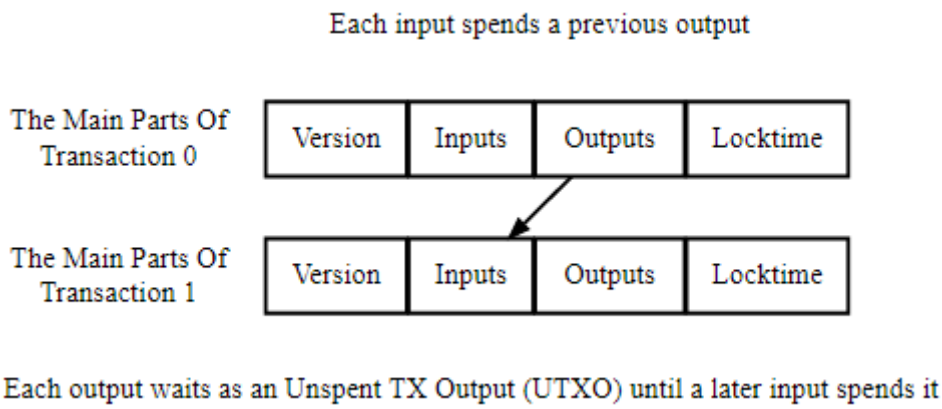
TXID

TXID (Transaction Identifier) Explained

By **Bisade Asolo** - October 29, 2018

TXID (Transaction Identifier) Explained

Transactions on networks such as Bitcoin operate using the UTXO (Unspent Transaction Output) model. With this model, nodes on the network track all available spendable transaction outputs, also known as unspent transaction outputs or UTXOs. These unspent transaction outputs are then used as inputs in the formation of new transactions on the network. TXID, also known as a transaction identifier or a transaction hash, is a unique piece of data that is used to identify a transaction.



Source: Bitcoin.org

As the image above shows, each transaction possesses at least one input and one output, with each input spending bitcoins contained in a previous output. The output, now a UTXO, sits in the UTXO set until it is spent by a later input. Thus, when a user's wallet indicates a spendable balance of 5 bitcoins, this can be interpreted to mean that the user has 5 bitcoins waiting in one or more UTXO.

Transactions are broadcasted in a serialized byte format known as the raw transaction format. In computer science, serialization is the process of translating the internal representation of a data structure into a format that can be transmitted one byte at a time. This raw transaction format is then hashed twice using a cryptographic hashing algorithm, which in the case of Bitcoin is SHA-256, to produce the TXID of a transaction.

TXID & Segregated Witness (SegWit)

The TXID is not guaranteed until a transaction has been confirmed by the network. However, payment tracking by use of the TXID can be affected by transaction malleability. This occurs when the TXID of a transaction is altered before it can be confirmed in a block. This is problematic as it creates opportunities for attack against poorly coded wallet software that assume unconfirmed TXIDs are immutable. With Bitcoin, the introduction of Segregated Witness or SegWit, was designed to serve as a solution to transaction malleability.

SegWit is an upgrade to the Bitcoin consensus rules and network protocol that was proposed and implemented as a BIP-141 soft fork. In the field of cryptography, the term witness is used to describe a solution to a cryptographic puzzle. With Bitcoin, the witness serves as a solution to the puzzle placed on a UTXO. This cryptographic puzzle is also known as a locking script, a witness script or a scriptPubKey, and it determines the conditions that must first be satisfied before a UTXO can be spent. In the context of the Bitcoin protocol, a digital signature is one type of witness that can be presented to solve the cryptographic puzzle in order to spend funds. However, the term witness can be more broadly thought of as any solution that can satisfy the conditions imposed on a UTXO. In the case of Bitcoin, the term witness can be thought of as being a more general term for the unlocking script, which is also known as signature script or scriptSig. The unlocking script is a collection of data parameters that are generated by a spender to satisfy the conditions placed by the locking script.

When a transaction consumes (spends) a UTXO, it must provide a witness. The locking script attached to a UTXO requires that the witness data be provided in the input part of the transaction that consumes the UTXO. However, with a SegWit UTXO model, a locking script can be satisfied with the witness data outside of the input (segregated). By moving the witness outside of a transaction, the TXID used as an identifier for a transaction no longer includes the witness data. As the witness data is the only element of a transaction that can be modified by a third party, separating the two also eliminates the opportunity for transaction malleability attacks.

With the implementation of SegWit, transactions on the Bitcoin network now have two identifiers, TXID and WTXID. TXID is the traditional transaction ID, which is the double SHA-256 hash of a serialized transaction without the witness data. Whilst the new transaction of WTXID is the double SHA-256 hash of the new serialization format of a transaction with the witness data. Thus, since SegWit transactions do not contain witness data in every input, there is no part of the transaction that can be modified by a third party.