# **Schnorr Signatures Simply Explained. Schnorr VS ECDSA**

With the constant discussion and testing in the past few years in the Bitcoin community, about the implementation of Schnorr Signatures with the upcoming Taproot update, some people are wondering what are Schnorr Signatures, and how does it compare to Bitcoin's Elliptic Curve Digital Signature Algorithm (ECDSA).

## **What is ECDSA?**

Without going into the complicated math equations, ECDSA is a digital signature scheme that creates encrypted messages through an algorithm known as the Elliptic Curve, a method that Bitcoin uses to help people verify the ownership of their funds within their addresses, where a public key is used to broadcast a transaction that its private key authentically created. Using the public key, the broadcasted message or transaction can be verified to be genuine with it, verifying that it was created by the correct person because it can see it was signed by the private key and mathematically match up with it.

## **What is Schnorr Signature?**

A Schnorr Signature is, like ECDSA, a method of creating encrypted messages through its own algorithm, but the difference between it and ECDSA is that it has the ability to merged multiple messages, or transactions if we're talking about Bitcoin/cryptocurrency, and appear as if it was a single transaction being executed with a single signature between one public key and another. This would be known as signature aggregation.

## **Schnorr VS ECDSA**

It could definitely be argued that Schnorr Signatures is the next step after ECDSA as it maintains what it does, with the difference being that it adds an extra feature to its process that results in decreased file size and increased privacy.

Let's take an example. If we had a multisig wallet set up and we're using ECDSA, where we'd need the signature of 2 out of the 3 connected addresses. Once we have our two signatures to broadcast a transaction, the network will see the multisig set up with their public keys, all the input addresses and output address of the transaction.

Now if we've done the same thing but using Schnorr Signatures instead, the public keys used in the creation of the multisig setup will appear as just one address. When broadcasting a transaction, all the public addresses of outputs and inputs would be combined to make it appear that there's just a normal transaction between two individuals with a total of just two public keys, one sender and one receiver.

Because of this, your privacy is increased as you send out normal transactions, as well as when you're dealing with complicated transactions such as what's being done with multisig wallets, all the while you're paying less transactions fees because you're now sending out less data for the blockchain network to handle, which also frees up space within it, resulting in even lower fees, as well as more space within each block I the blockchain to fit more transactions.

All in all, Schnorr Signatures' signature aggregation better hides your public key/s, in comparison to what ECDSA exposes, and because of its method of concealing such data, it resulted in decreased transaction fees alongside lessening the congestion of the blockchain.