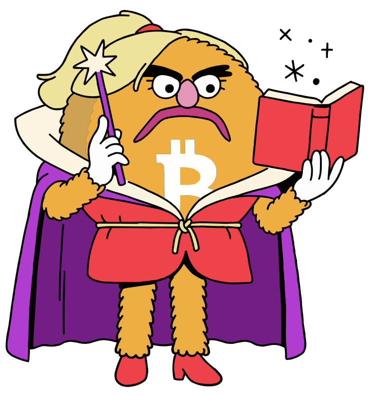
## *Thanks for reading!*

That was just a brief overview of Taproot. Hope you enjoted it! Ready to learn more? Want to print free copies of this zine? Visit



**https://satsie.dev/zines**

for additional resources and more content like this!

satsie's pocket guide to

# *TAPROOT*

A SHORT OVERVIEW OF BITCOIN'S TAPROOT UPGRADE

The Taproot Upgrade has 3 parts

- BIP-34Ø: Schnorr
- BIP-341: MAST + Taproot
- BIP-342: Tapscript

When Taproot is discussed in a general sense, including how we've talked about it so far, it is usually in reference to all 3 of these things.

Let's take a closer to look at each!

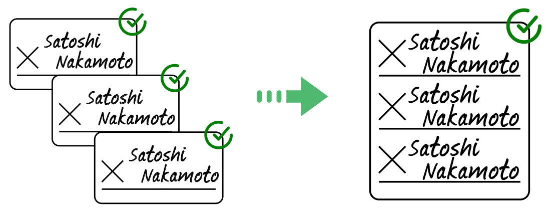
### ☆ BIP-34Ø: Schnorr ☆

This BIP introduces Schnorr, a new signature scheme.

Compared with ECDSA, the other signature scheme ***bitcoin*** uses, Schnorr signatures are more secure, easier to work with, and slightly more efficient.

### ☆ Batch signature validation ☆

Validating digital signatures usually requires a lot of efforts from a computer's CPU. Now transaction signatures can be grouped together and validated as one unit, instead of one by one.



### ☆ Better privacy while spending ☆

***bitcoin*** allows you to specify multiple ways to spend a coin. Prior to Taproot, all these ways had to be made public when the coin was spent. This is bad for privacy, especially for coins with unique spending rules, making them easy to identify.

## What is the Taproot Upgrade?

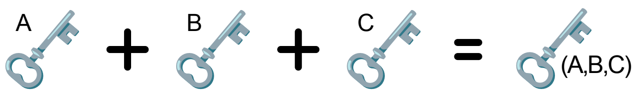
Taproot is a set of improvements that allow ***bitcoin*** to be used in a more scalable and private ways.

Activation date: November 2021
Block height: 709,632.

## Taproot enables some cool features

### ☆ Key and signature aggregation (MuSig) ☆

If you have public keys A, B and C, they can be combined into one. The same is true for the corresponding signatures.



This means complex multisignature spends can look like ones that only involves 1 key.

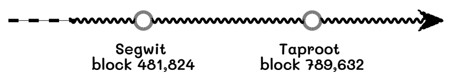
With Taproot, the only thing that needs to be public is the specific way in which a coin was spent, not all other possibilities. This means:

1. less data on the blockchain, and
2. more privacy!

**Together, many features contribute to what is perhaps Taproot's most impressive use case: making many different ways of spending *bitcoin* indistinguishable from one another. It doesn't matter how simple or complex the spending rules are.**

## Taproot vs. SegWit

SegWit is the major upgrade that came before Taproot.



It contains many things, including a new version field to use with transaction output scripts

For SegWit this value is set to "Ø" ("SegWit vØ"). For Taproot, it's set to "1". This is why you'll sometimes see Taproot scripts referred to as "SegWit V1".

SegWit and Taproot are separate upgrades that result in different transaction output types. Taproot builds on the foundation SegWit created.

## The Taproot Upgrade was a Soft Fork

This means the upgrade narrows consensus rules, or constrains the rules of the system. Soft forks require majority hashpower from miners but the upgrade is optional for everyone else. Any behavior that was invalid before continues to be invalid, and nodes running older version of ***bitcoin*** continue to be compatible with newer versions.

### ☆ BIP-341: Script Trees + Taproot ☆

This BIP is made of 2 thigs:

1. **Script trees:** Tree-like data structures used to compactly encode multiple scripts. In this BIP, each leaf represents a single script and only one leaf may be chosen by the spender. The spender is responsible for showing the path of the leaf (AKA the "Merkle branch").
2. **Taproot:** A technique that allows a coin to be spent by public key OR by script. Taproot leverages the power of MAST and Schnorr to make transactions more **flexible, private and efficient.** With Taproot, you can set up many different spending constraints, but only reveal the one that is used!

### ☆ BIP-342: Tapscript ☆

Script is the ~~terribly uncreative~~ name for ***bitcoin***'s smart contract language. Tapscript is an upgraded scripting language that supports Schnorr and Taproot.