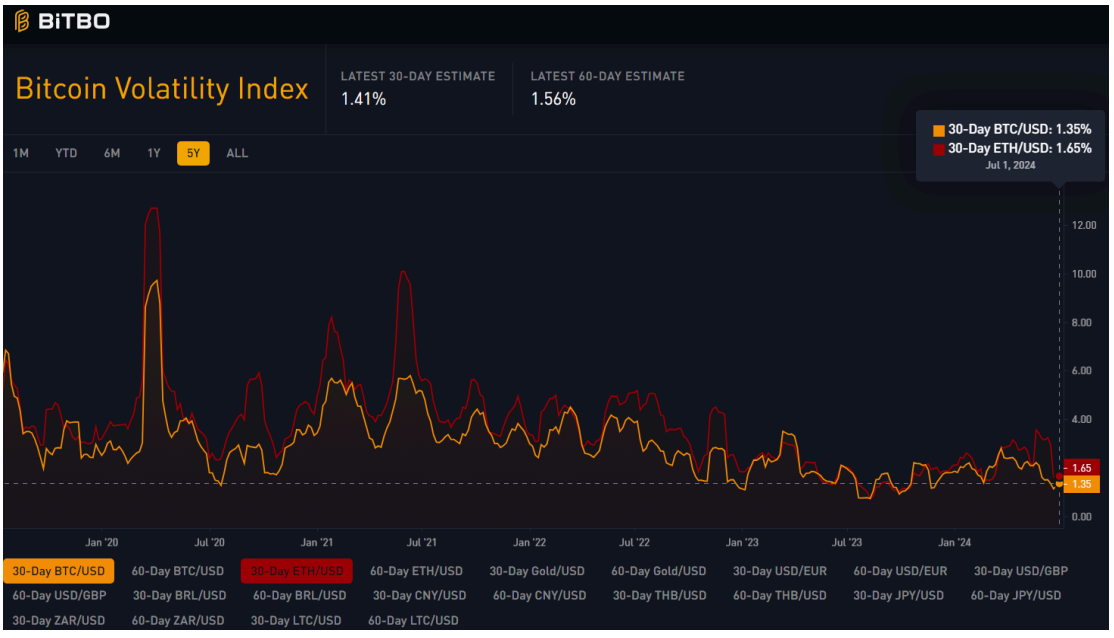
BTC Staking Part Two: Restake via Babylon

Exploring the Role of Babylon

Babylon's restaking works by locking BTC to the Bitcoin mainnet and then providing shared security for PoS consumer chains. Restaking means you can earn yields with your yield-bearing staked funds again. In EigenLayer, restaked ETH can earn both AVS and ETH2 PoS rewards (we don't consider potential airdrops here). Since yields are only generated when providing security to PoS chains by locked BTC, we think Babylon can also be viewed as a part of the BTC Staking narrative.

As EigenLayer already provides a shared security marketplace, what value proposition does Babylon add? These two protocols have the same position, but the key difference here is the underlying assets. Crypto prices are like a roller coaster, especially in this cycle. One of the common points of ETH and BTC is that they are less volatile than Altcoins. According to the chart below, the 30-day BTC price is less volatile compared to ETH. Besides, the altcoins market cap is highly affected by BTC. Therefore, BTC is the most secure asset when providing economic security.



Source: https://bitbo.io/volatility/

The second difference is the staking ratio between BTC and ETH. According to Staking Rewards, about 27.32% of ETH are staked. However, BTC is not PoS-based and cannot earn yields natively. Although many BTC L2 are trying to fill that gap, most of the BTC stays idle. Babylon brings restaking, Liquid Restaking Tokens (LRT), and Liquid Staking Derivatives (LSD) narratives to Bitcoin. These narratives are valued at 77B in Ethereum now as per DefiLlama.

The third difference is the ecosystem. Currently, Babylon supports the Cosmos SDK-based ecosystem while EigenLayer supports Ethereum. While ATOM has higher volatility compared to ETH, introducing BTC to Cosmos can help Babylon gain adoption from the Cosmos ecosystem.
Based on the above differences, Babylon has opened a new chapter in improving the capital efficiency of Bitcoin.

Babylon: Making Bitcoin Capital Efficient

Babylon helps make Bitcoin more capital efficient by allowing BTC staking while maintaining security by adding timestamping PoS block hashes and votes of validators on Bitcoin.

Bitcoin Timestamping

PoS-based chains will face a long-range attack: malicious attackers can attack PoS chains by buying exited validator keys and then making fake blockchain history. Current PoS chains prevent such attacks by social consensus, which means they publish blockchain history periodically. Yet it adds a trust assumption that the entity associated with the chain will not publish any false history.

Babylon timestamp PoS block hashes and votes of validators on Bitcoin instead of social consensus to eliminate additional trust. **It also provides shorter unbounding time and censorship resistance.** Babylon also offers checkpoint aggregation and data availability services. In addition, via CosmWasm, one can develop timestamping services with any data verification rules.

Bitcoin Staking

Compared to current BTC L2 solutions, **staking via Babylon does not need any bridging**, thus there are no bridge hacking risks. In addition, **Babylon staking is non-custodial**,

which means staker locked their BTC in the Bitcoin mainnet instead of other sidechains or rollups. Such an approach preserves the security of BTC holdings while enabling users to earn rewards from Altcoin PoS validation. Babylon's BTC Staking innovation not only ensures security and accessibility but also redefines capital efficiency in the evolving BTC Staking landscape.

Roles in Babylon

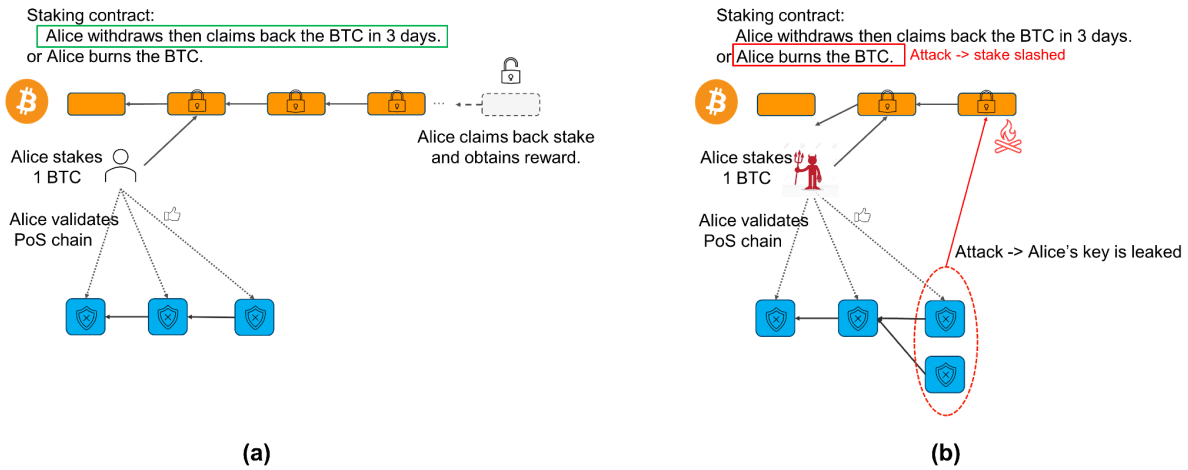
Staker

Since there are no Turing-complete smart contracts on Bitcoin, staking contracts must be expressed as UTXO transactions written in Bitcoin scripts. A staking contract is an example of a "Bitcoin Covenant", which restricts how the output of a transaction can be spent. Covenants can be implemented via **OP_CHECKTEMPLATEWERIFY**, a set of Bitcoin scripting language opcodes proposed for inclusion in the next upgrade of the Bitcoin scripting language. Babylon simulates convenant in a trustless way by following procedures:

A staker (Alice) enters into a staking contract by sending a staking transaction to the Bitcoin network, locking the bitcoins in a self-custodial vault. The staking transaction's input is the address of the staker's BTC and the output can be used in **Unbonding transaction or Slashing transaction**:

1. The staker sends an unbonding transaction, BTC will be unlocked and returned to Alice within 3 days.
2. The staker sends a slashing transaction, which sends the BTC to the burned address (unspendable output).

Once this staking transaction appears in the Bitcoin, Alice's stake can start earning rewards by validating for PoS chains. During her validation duties, there are two possible paths:



Source: https://docs.babylonchain.io/papers/btc_staking_litepaper(EN).pdf

1. **Normal One**: Staker follows the protocol honestly, and when she wants to unstake, she sends an unbond transaction to the Bitcoin. Once the unbond transaction enters the Bitcoin chain, the validation duties stop at the PoS chain. After 3 days, the withdrawal request is approved and BTC is returned to Alice. The PoS chain will also grant Alice altcoin rewards.
2. **Malicious One**: The second type means staker involved in double-spending attack of the PoS chain. In this case, the staker's private key is leaked to the public. Now anyone can send a slashing transaction to the Bitcoin network and burn staker's BTC.

Validator

Babylon's validators have the same properties as other cometBFT chains developed by Cosmos SDK. Validators are responsible for submitting new blocks in the chain. These validators participate in the consensus protocol by broadcasting votes that contain cryptographic signatures signed by each validator's private key.

Finality Provider

Finality Providers are responsible for consensus security. Generally speaking, PoS protocols penalize double signing and surround voting. Babylon chooses to implement slashing using accountable assertions EOTS and finality gadget.

Extractable one-time signatures（EOTS）refer to the fact that when a signer signs two messages with the same private key, the private key is compromised, i.e., the private key can be extracted from the signatures of two messages. EOTS has been proposed as a generalized method to penalize ambiguity, e.g., double spending the same BTC outputs. But although EOTS can handle it, it cannot do anything about surround voting.

As Consensys' report said:

In Ethereum, ***surround vote*** *is a validator casting an FFG vote that surrounds or is surrounded by a previous FFG vote they made. Here are two examples based on a scenario that a validator made an FFG vote in Epoch 5 with a source of Slot 32 and a target of Slot 128:*

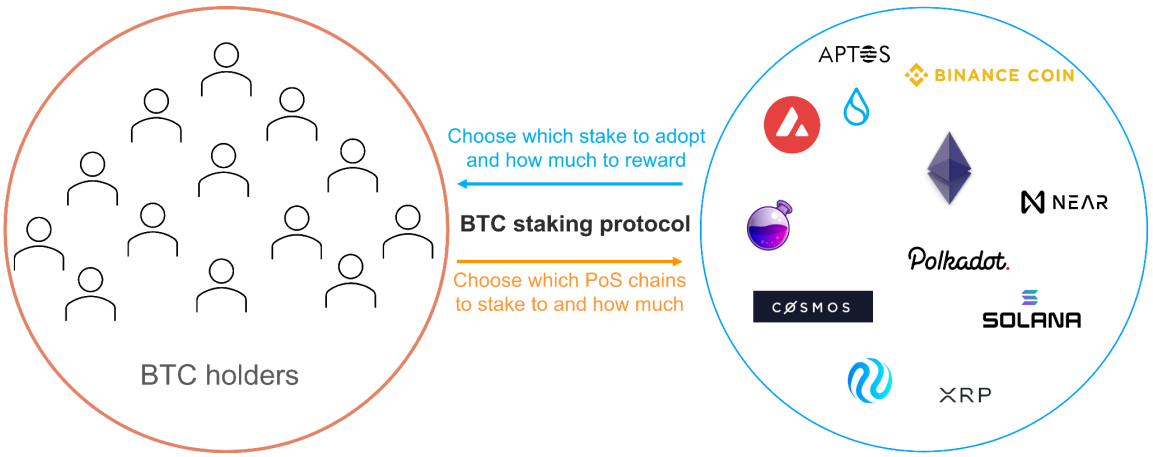
- ***surrounded by a previous FFG vote****: An FFG vote in Epoch 6 with a source of Slot 64 and a target of Slot 96, would be an FFG vote that was* ***surrounded by*** *their Epoch 5 vote.*
- ***surrounded a previous FFG vote****: An FFG vote in Epoch 6 with a source of Slot 0 and a target of Slot 160 would* ***surround*** *their FFG vote in Epoch 5.*

Babylon deals with such a case by adding a round of Extractable One Time Signatures (EOTS) after cometBFT finalizes a block(We will describe EOTS in the subsequent section). A block is considered truly final if it is finalized by both cometBFT and receives EOTS signature by more than 2/3 of the finality providers. This additional round of signatures is an EOTS finality gadget.

If there is a security violation in this modified protocol, more than 1/3 of the finality providers will use EOTS to sign two blocks at the same height. This can lead to the extraction of the private keys of these involved parties. In addition, the EOTS signing scheme can be implemented through Schnorr signatures.

PoS Consumer Chains

PoS consumer chains are those newly launched PoS chains that do not have sufficient economic security. They pay their native asset rewards to purchase the BTC-shared security service of Babylon.



Source: https://docs.babylonchain.io/papers/btc_staking_litepaper(EN).pdf

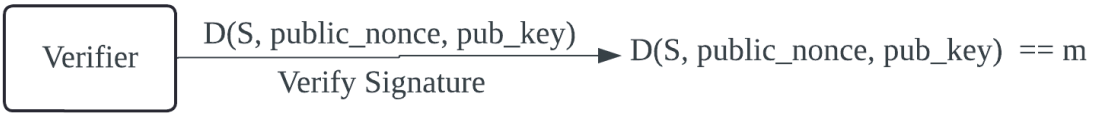
Dive into EOTS

To understand Schnorr signature-based EOTS, we need to understand some of the fundamentals of Schnorr signature first.

In the Schnorr signature algorithm, the signer has key pair **K = (secret_key, public_key)**. To sign a new message m, the signer performs the following steps:

1. Generates a **secret_nonce**
2. Use the **secret_key** and the **secret_nonce** to sign the message.

3. Computes the corresponding **public_nonce** of the **secret_nonce**
4. Publish the signature **S** and **public_nonce** to allow the public to verify **S**

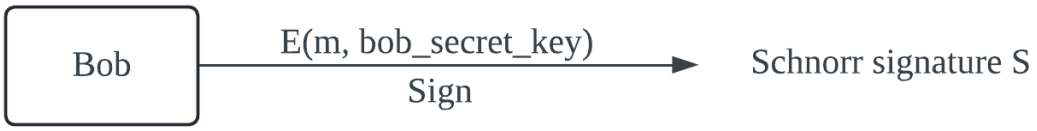


Here, the **secret_nonce** must be a new one for every new message. If the same **secret_nonce** is used to sign two different messages, then anyone who observes the two signatures can extract the signer's **secret_key**.

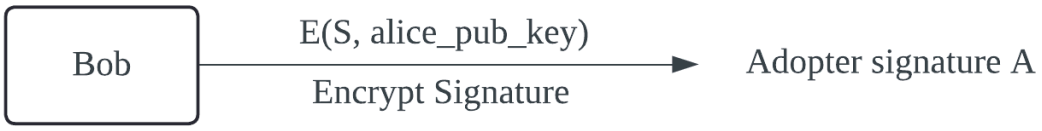
EOTS is a scheme derived from Schnorr signatures and works as the following picture:

Source:https://medium.com/babylonchain-io/technical-preliminaries-of-bitcoin-staking-74a42b283d79

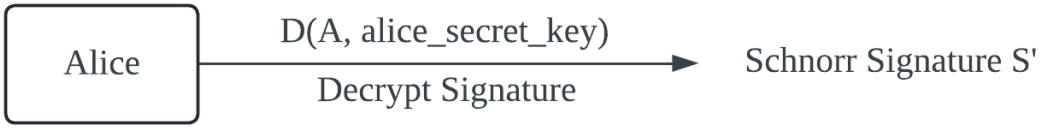
1. Assume Bob wants to send Alice 4 BTC, Bob signed a message (Bitcoin tx) **m** and get schnorr signature **S**



2. Bob encrypted **S** with Alice's public key **alice_pub_key** and published adopter signature **A**

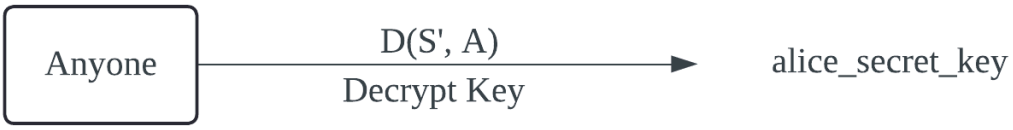


3. **Alice uses A and alice_secret_key** can recover Schnorr signature **S'**: **S'** is different from S, but it's still a valid Schnorr signature from Bob. It's like Bob uses two nonces to produce two different signatures **S** and **S'**.



4. Alice submits Bitcoin tx **M**, **A and S'** to proceed with the transaction.
5. **Schnorr signature S' and A** allow anyone to decrypt Alice's secret key **alice_secret_key**.

   In other words, once Alice submits S' and A to the Bitcoin network, her secret key can be computed by anyone, and all of her funds can be transferred.



Potential Risks of Babylon

Stakers have the risk of being slashed, but it will only happen if they act maliciously. Also as newly launched PoS asset prices are highly volatile, which may cause staker to unstake their BTC thus decreasing the supply of economic security of PoS consumer chains.

Conclusion

Babylon offers a shared security marketplace with the least volatile crypto asset BTC. With Babylon's case, we've further refined our BTC Staking thesis. The future of BTC Staking belongs to protocols that fulfill the following three properties: Non-Custodial BTC Staking; Security, and Maximize Yields.

Reference

1. https://medium.com/babylonchain-io/technical-preliminaries-of-bitcoin-staking-74a42b283d79
2. https://messari.io/report/babylon-bitcoin-shared-security-and-staking
3. https://mirror.xyz/chakrabtc.eth/fjj4yVd3rA380Rx2EnZUo5Ww-bfYmaQZPe4CJFRpVk8
4. https://consensys.io/blog/the-ethereum-2-0-beacon-chain-explained

Disclaimer

investors. Any forward-looking statements in this material are subject to several conditions, uncertainties and assumptions. We undertake no obligation to update or revise any forward-looking statements.