Bitlayer: A Bitcoin Computational Layer Architecture Based On BitVM Paradigm

Bitlayer Research Group

build@bitlayer.org

April 5, 2024

Abstract. Bitcoin, as the blockchain with the highest value, strongest security, and highest decentralization, still faces challenges in scalability and the limitation of smart contracts, greatly limiting its ecosystem's development. Existing solutions have not yet to totally address these issues. Bitlayer proposes a Turing-complete computing layer solution based on the BitVM paradigm and zero-knowledge(ZK) proofs, while fully inheriting Bitcoin's security. In Bitlayer, we introduce the Layered Virtual Machine(LVM), which decouples the front-end smart contract execution layer from the back-end ZK Generator. Additionally, Bitlayer combines Bitcoin's Taproot technology with FRI (Fast Reed-Solomon IOP of Proximity), aiming to achieve the most efficient ZK-STARK Verifier currently available on Bitcoin. Furthermore, Bitlayer will implement a trustless dual-channel cross-chain bridge based on BitVM and OP-DLC(Optimistic-Discreet Log Contract) technology.

Keywords: BitVM, Bitcoin Friendly FRI, OP-DLC, Layered Virtual Machine

1 Introduction

Bitcoin, as the first successfully implemented cryptocurrency, has attracted global attention and continuous exploration since its inception. However, despite Bitcoin's advantages such as decentralization and high security, issues related to transaction speed and costs on its main chain have been a focal point of industry concern. Additionally, the limitations of Bitcoin's smart contracts have restricted the development of applications within the Bitcoin ecosystem. In order to address these issues, Bitcoin Layer 2 solutions have emerged, gradually becoming one of the hottest topics in the crypto industry.

Currently, in order to improve the transaction throughput and efficiency of the Bitcoin network, scalability technologies such as sidechains and state channels have emerged. Among them, state channels provide secure and diverse payment solutions with advantages in real-time processing, privacy protection, and cost-effectiveness. However, their interaction primarily involves two parties, and each interaction requires complex processes to open and close channels. Additionally, state channels have limited capabilities in verifying arbitrarily complex

computations. This limitation reduces their applicability in scenarios requiring complex, conditional logic, and interactions.

On the other hand, while sidechain solutions offer stronger independence, scalability, and flexibility in design, and even provide better smart contract support than Bitcoin, they do not inherit Bitcoin's robust security. This security difference stems from sidechains employing independent consensus mechanisms, which are far less robust compared to Bitcoin's consensus mechanism.

So we propose Bitlayer, a Turing-complete Layer 2 solution on Bitcoin, with equivalent security inherited. In Bitlayer, we built a layered virtual machine technology derived from the BitVM solutions. This technology supports arbitrary types of computational operations through zero-knowledge proofs and optimistic execution mechanisms, and verifies the validity of computations on Bitcoin. The layered virtual machine technology allows us to support any frontend types of smart contracts such as EVM and backend proof generators such as ZK-STARKs/ZK-SNARKs, while ensuring security and maximizing computational flexibility. Additionally, Bitlayer combines BitVM with OP-DLC (Discreet Log Contracts with Optimistic mechanism) technology to build a secure and decentralized assets bridge mechanism. This solution bridges the assets between Bitlayer and Bitcoin, introducing a forced withdrawal mechanism to ensure asset security under any circumstances, even after Bitlayer vanished.

2 Architecture

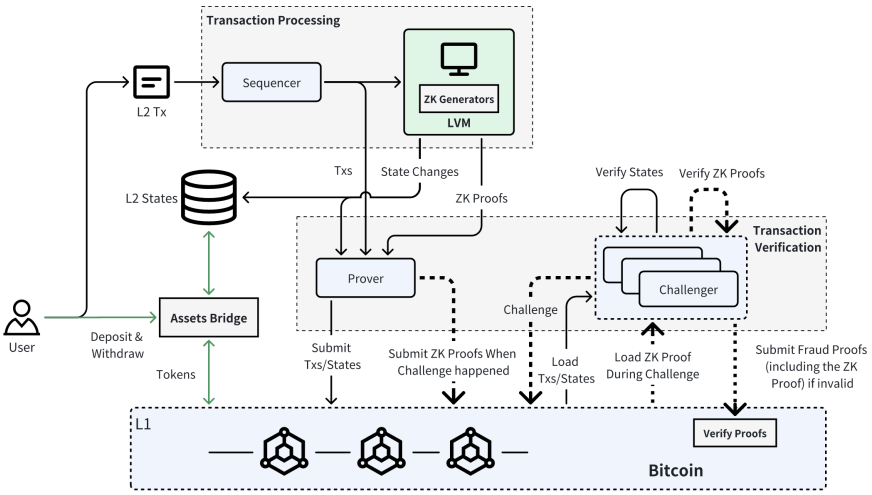


Fig. 1. The Architecture of Bitlayer

Bitlayer has revolutionized the verification process for Layer 2 transactions using optimistic execution, while keeping the Bitcoin protocol intact. Its architecture comprises transaction processing, verification, and asset bridge components. Transaction processing involves a sequencer and layered virtual machine(LVM), optimizing transaction handling and computational efficiency. Transaction verification, managed by provers and challengers, ensures transaction validity and compliance with network rules. They collaborate to complete the entire process from Layer 2 transaction handling to Layer 1 confirmation, maintaining transaction security and integrity throughout the process. Bitlayer's assets bridge components further enhance its capabilities by enabling interoperability between Layer 2 and Layer 1 networks, facilitating secure asset transfer across blockchain layers. The overall Bitlayer's architecture can be demonstrated as follows.

2.1 Transaction Processing

Transaction Processing, as illustrated in the figure above, involves the sequencer and layered virtual machine. These components are responsible for the entire transaction handling, starting from transaction acceptance to executing the output.

Sequencer: Like other Layer 2 solutions, the sequencer in Bitlayer is responsible for collecting cached transactions and sorting them, serving as the entry point for transactions in Bitlayer.

Layered Virtual Machine(LVM): The LVM is the computing component of Bitlayer, responsible for executing smart contracts and generating the latest states and zero-knowledge proof. Challengers then use this proof to challenge the execution results.

2.2 Transaction Verification

In Bitlayer, transaction verification is achieved by a zero-knowledge-based optimistic mechanism between the prover and challenger.

Prover: The Prover is responsible for submitting Layer 2 transactions, and states of execution to the Layer 1 chain as described above. It also reveals zero-knowledge proofs on the chain when getting challenged.

Challenger: The Challenger is responsible for verifying the execution results submitted by the Prover through states of execution and zero-knowledge proof verification. If malicious behavior is detected, the Challenger initiates a challenge process to generate fraud proofs including invalid zero-knowledge proofs, and submits them to the Layer 1 chain.

2.3 Assets Bridge

The Bridge acts as a crucial component in Bitlayer's infrastructure, facilitating the seamless movement of assets between Layer 2 and Layer 1. Its primary

responsibility is to ensure the secure transfer of user assets through an innovative combination of OP-DLC and BitVM.

3 Bitlayer Execution Protocol

In Bitlayer, we proposed an execution mechanism derived from BitVM with zero-knowledge proof and an optimistic mechanism, which achieves off-chain execution of transactions and on-chain verification of Bitcoin without the requirements to modify the underlying Bitcoin protocol. This solution implements a Turing-complete Bitcoin Layer 2 smart contract solution, enabling verification of any complex off-chain computation on the Bitcoin blockchain. Additionally, we also proposed a compound virtual machine model called Layered Virtual Machine (LVM), which decouples the supported types of smart contracts from backend zero-knowledge proof verifies, enabling independent optimization and extension. Bitlayer significantly broadens Bitcoin's potential applications, allowing it to serve not only as a means of payment but also as a validation platform for various decentralized applications and complex computing tasks.

3.1 Optimistic Execution With Zero-Knowledge Proof

Bitlayer employs the same optimistic execution mechanism as Optimistic Rollup, which assumes all received transaction batches are valid unless a participant can prove the invalidity of any transaction in that batch. On the one hand, assuming transactions are valid allows for the fastest on-chain processing; on the other hand, the transaction will be reverted as long as one participant can honestly prove its invalidity, which adopts a 1-of-n security model. Additionally, Bitlayer applies zero-knowledge to reduce the cost of the challenge. Through this approach, the optimistic execution mechanism increases transaction throughput, lowers costs, and maintains the security and decentralization properties of the underlying blockchain.

3.1.1 Processing Framework

In Bitlayer, the optimistic execution mechanism consists of processes such as transaction batching, transaction execution with zero-knowledge proof generation, transaction and state submission with zero-knowledge proofs, challenges, and state rollbacks. In the optimistic execution mechanism, transactions are initially batch-processed off-chain and then submitted with states and zero-knowledge proofs to the Layer 1 blockchain in a compressed format, reducing data load and associated costs. Once processed, other participants have a challenging period to submit fraud proofs if incorrect transactions or invalid state transitions are detected by zero-knowledge proof verification. Successfully accepted fraud proofs will trigger state rollbacks and penalties for malicious actors, which ensures the security and integrity of the whole system.

3.1.2 Data Availability

The challenge process in the optimistic execution mechanism relies on the transaction submitter's disclosure of transactions and the current state. Therefore, Bitlayer requires a secure, highly reliable, and decentralized data availability mechanism. Currently, Bitlayer utilizes Bitcoin to provide data availability because Bitcoin is the most robust, secure, and decentralized blockchain in terms of consensus.

To meet Bitcoin's on-chain requirements, submitted data is compressed and split into multiple transactions submitted to Bitcoin, organized like Bitcoin inscriptions. Subsequently, validators filter transactions, identify all inscriptions and reconstruct the original data.

In the future, to reduce operational costs, Bitlayer is also researching integrating other data availability modules such as Celestia, Data Availability (DA) Committee, etc., while ensuring the same level of security, reliability, and decentralization.

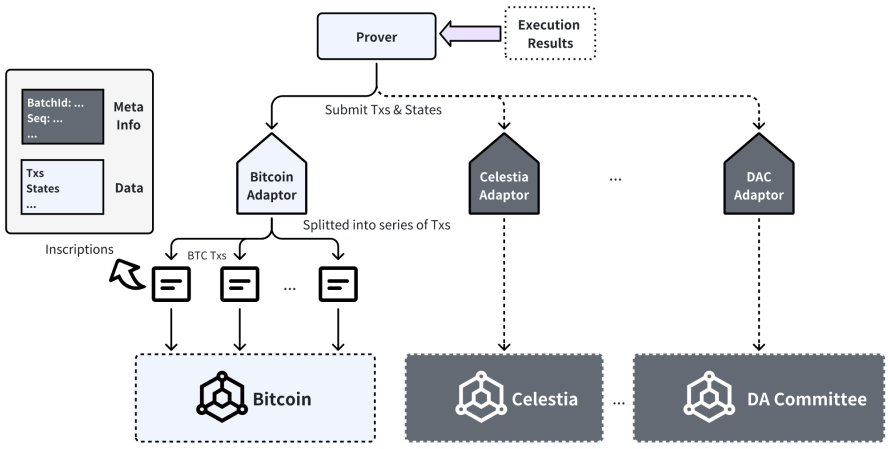


Fig. 2. Data Availability in Bitlayer

3.1.3 Rollup Proof Generation

The illustration below delineates the rollup-proof generation of the Bitlayer protocol, comprising five distinct circuits:

– **Execution Circuit:** This circuit processes state checkpoints and user transactions, encodes the Bitlayer state transition logic, and yields proof for verifying the accuracy of state transitions.

- **Aggregation Circuit:** It accepts multiple state transition proofs as input, encapsulating the logic for zk-proof verification, and produces a consolidated batch proof.
- **Bitlayer Block Derivation Circuit:** This circuit is fed with the Bitcoin canonical chain, represented by the finalized block hash, encoding the logic for Bitlayer block derivation, and yielding the derivation proof.
- **Bitcoin Block Finalization Circuit:** Accepting the Bitcoin finalized block hash as input, this circuit encodes the Bitcoin PoW logic and finalization logic, generating the finalization proof. The Bitcoin finalized block hash is referred to as the trust anchor.
- **Rollup Circuit:** Taking proofs from the aforementioned circuits as input, this circuit encodes the linkage logic for the public inputs of these circuits, culminating in the generation of the final proof, which can be challenged using BitVM.

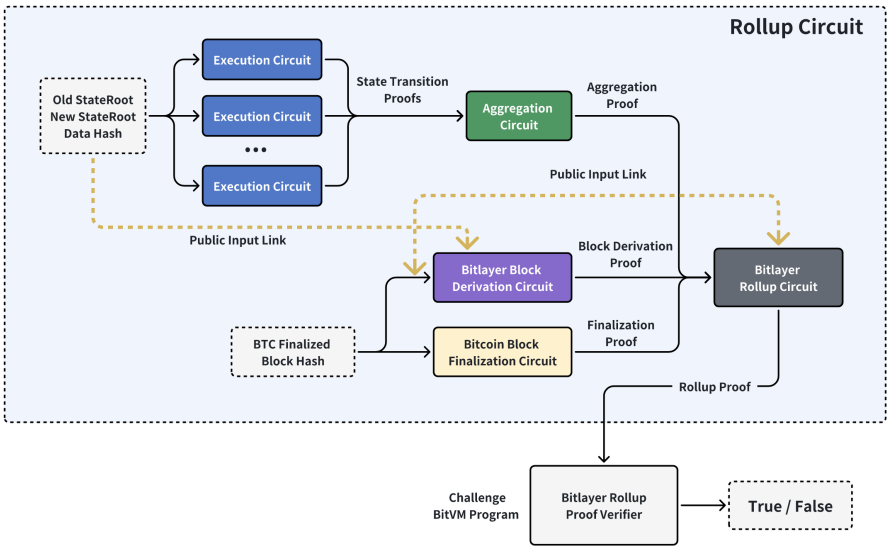


Fig. 3. Bitlayer Rollup Circuit

This design also fosters flexibility within the DA layer. Through adjustments to the block derivation circuit and the DA layer finalization circuit, Bitlayer can easily extend support to other DA layers. Additionally, we can incorporate a censorship resistance mechanism into block derivation logic to deter the Bitlayer sequencer from omitting user transactions in Layer 2 blocks.

3.2 Challenge Protocol

In an early version of BitVM(BitVM1), the challenger constantly uses the binary method to challenge the state of a specific instruction set to find the instruction where the Prover made a mistake. In the end, the Prover needs to use its declared state to execute the corresponding instruction. At this point, the Prover obviously cannot be successful, so it cannot retrieve its staked Bitcoin, while the Challenger can unlock the staked amount after the Timelock ends.

In the latest version of BitVM(BitVM2), unlike BitVM1, the Prover commits the entire program in segments to the leaf nodes all at once. Anyone can spend from any of these scripts exactly if an assertion $f_i(z_{(i-1)}) = z_i$ doesn't hold.

Bitlayer challenge game continues to use the BitVM2 challenge scheme, but we challenge the correctness of a STARK verifier's execution. We propose a Bitcoin-Friendly FRI to ensure that the FRI Verifier doesn't need to simulate the process of verifying the MerklePath in Tapscript. Instead, it directly utilizes the characteristics of the Taptree, committing all polynomial evaluations to the corresponding Taptree leaf nodes.

3.2.1 Commitments

Commitments are the foundations of data processing in the challenge protocol. Similar to BitVM, a series of commitments are defined to describe the computing process, including bit commitment, gate commitment, and circuit commitment, which all can be implemented into Bitcoin scripts.

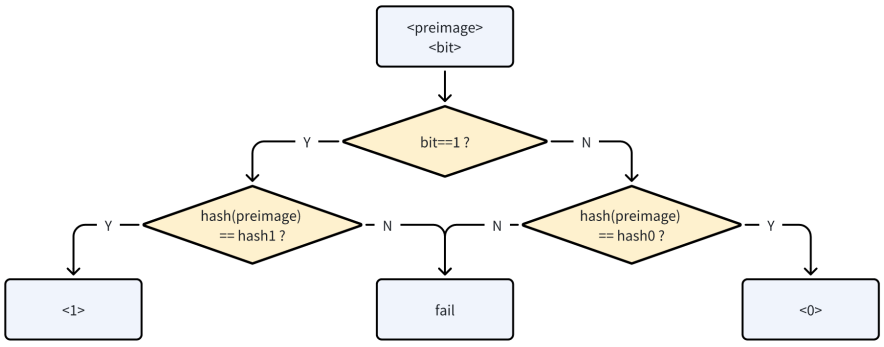


Fig. 4. Process of Bit Commitment Script

Bit Commitment: The concept of bit commitment is a variation derived from Lamport signatures. It involves two hashes, hash0 and hash1(and the corresponding preimages are preimage0 and preimage1). The prover then can determine the value of the bit: either "0" by disclosing preimage0, which corresponds to hash0,

or "1" by disclosing preimage1, corresponding to hash1. Bit commitments enable the prover to set a variable's value across various scripts and UTXOs which extends the execution runtime of Bitcoin's VM through transaction splitting across multiple transactions.

Custom Gate Commitment: Gate commitment refers to the commitment of a custom gate, which can be assembled to represent any computable function. It can be composed of any input and output which is represented by bit commitment and any Bitcoin script. The following figure demonstrates the NADN gate commitment.

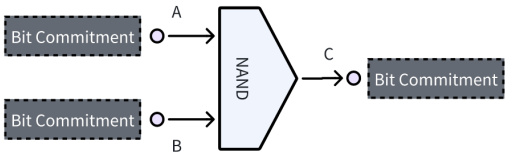


Fig. 5. NAND Gate Commitment

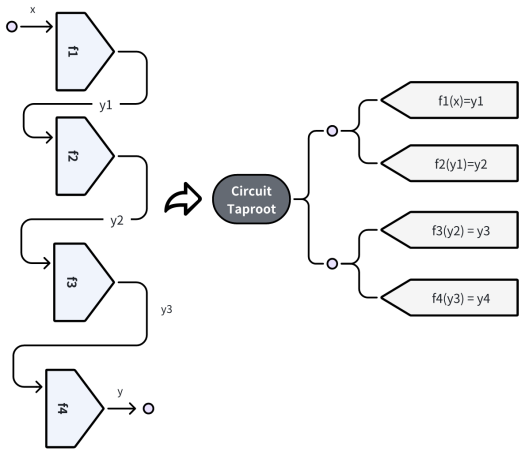


Fig. 6. Circuit Commitment

Circuit Commitment: Bit Commitment also has another name, called Copy Constraint. Once we have any gate constraints and copy constraints, we can ex-

press any computation with them. (zkVM has already proved this.) For example, we have a huge program f(x) = y, we can break it down into:

$$f_1(x) = y_1, f_2(y1) = y_2, f_3(y_2) = y_3, f_4(y_3) = y_4.$$

In this way, we need to design 4 custom gates: $[f_1, f_2, f_3, f_4]$. Therefore, we can distribute complex computations across multiple leaf nodes, thus enabling Bitcoin to implement ZK Verifier capabilities.

3.2.2 Challenge Game

Our ultimate plan is to hope for a validity proof in Bitcoin, but since the number of intermediate states of our optimized STARK Verifier is still relatively large, we continue to adopt an optimistic execution approach. The specific plan is as follows:

Prover Claim Correct State

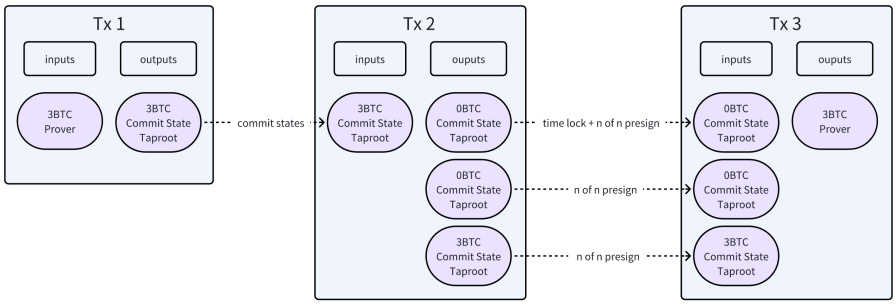


Fig. 7. Prover Claim Correct State

Bet Phase: Initially, the Prover sends 3BTC to the reveal-state-addr through Transaction 1 (Tx1).
Claim State Phase: The Prover reveals the starting and ending states of their claimed program by unveiling the corresponding Bit Commitment. It's important to note that Transaction 2 (Tx2) has divided a 3BTC input Unspent Transaction Output (UTXO) into three output UTXOs.
Waiting Finalize Phase: At this stage, the Prover waits for the Timelock's required time to expire before unlocking the transaction via the TimeLock. It's worth noting that unlocking this TimeLock transaction also requires signatures from n Operators.

In this case, Tx2 explicitly needs to divide one UTXO into three specific amount UTXOs. The Operator can calculate Tx2's hash before Tx2 happens, allowing them to know the specific content of Tx3's input UTXO. Simultaneously, Tx3's output to the Prover's Pay-to-Public-Key-Hash (P2PKH) address ensures

determinism. Hence, the Operator can pre-sign this transaction. Note that the SIGHASH here will use ALL, meaning it will apply to all inputs and outputs.

Furthermore, due to this pre-signature mechanism, if any of Tx2's output UTXOs are consumed, Tx3 cannot execute. Therefore, we can initiate challenges by consuming UTXOs.

Challenge Occurs

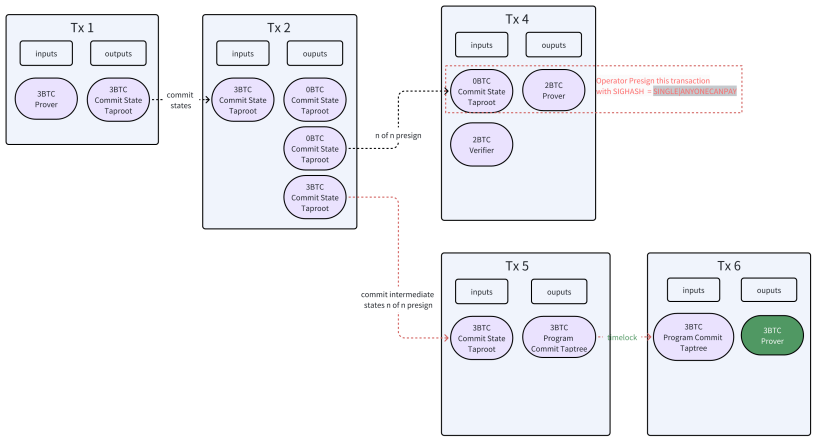


Fig. 8. Prover Claims Correct State and Challenge Occurs

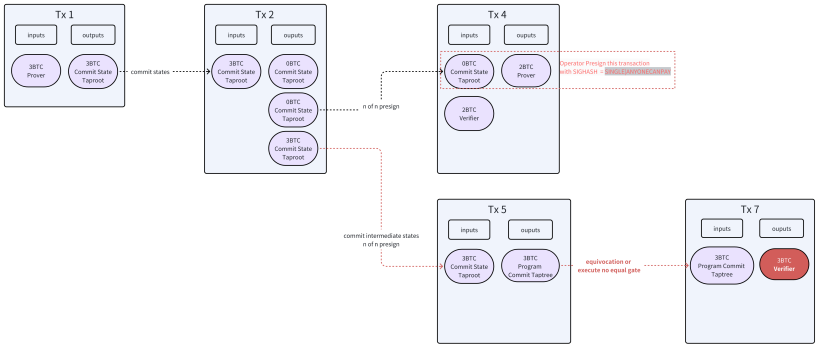


Fig. 9. Prover Claims Incorrect State and Challenge Occurs

In this scenario, the Bet Phase and Claim State Phase remain unchanged, but a challenge arises during the Waiting Finalize Phase.

Verifier Bet Phase: If anyone disputes the state claimed by the Prover, they can initiate a challenge with Transaction 4. This transaction consumes the UTXO required by Transaction 3, preventing its execution. Simultaneously, Transaction 4's SIGNHASH=SINGLE|ANYONECANPAY allows anyone to launch this challenge by deploying 2BTC.
Prover Reveals Intermediate States: In Transaction 5, the Prover consumes the 3BTC UTXO and discloses all intermediate states.
Verifier Wins the Challenge Phase: If the Verifier detects any issues with the intermediate states revealed by the Prover in Transaction 5, they can find the corresponding unlock script in the current Taptree. This allows them to win the challenge and earn a 1BTC reward (3-2).
Prover Wins the Challenge Phase: If all states provided by the Prover are accurate, the Verifier will not find the corresponding faulty gate logic. As a result, the Prover can claim their pledged BTC after the Timelock expires, winning the 2BTC initially bet by the Verifier to start the challenge.

3.2.3 Zero-Knowledge Proof Verification on Bitcoin

There are two primary categories of Zero-Knowledge Proof (ZK-Proof) systems:

- Elliptic curve pairing-based ZK-Proof systems, such as BN254-based Groth16 proof system.
- Hash function-based ZK-Proof systems, such as STARK and its variations.

The security of elliptic curve pairing-based ZK-Proof systems relies on the elliptic curve, typically with a prime field of around 256 bits. Due to Ethereum's implementation of the BN254 precompiled contract, the verification cost for ZK proofs based on BN254 curve pairing is economical. However, Bitcoin does not support the OP_MUL opcode, and implementing pairing operations would require a large script that exceeds the single standard block capacity limit. To realize a BN254 curve pairing-based ZK Verifier on Bitcoin, the following optimization strategies could be considered:

- Utilize schemes like Groth16, or FFlonk (where the FFlonk Verifier algorithm is simpler than Groth16's Verifier algorithm, albeit at the cost of increased computational load on the Prover).
- Customize a smaller parameter curve for Bitcoin, though this would reduce security, as attempted by the Liquid team, resulting in approximately only 50 bits of security.

One significant advantage of hash function-based ZK-Proof systems is their post-quantum security, in contrast to elliptic curve pairing-based ZK-Proof systems. Therefore, in the long term, STARK might be a more appropriate choice. Current STARK systems use smaller finite fields, such as M31 and BabyBear, which are Bitcoin-friendly. These fields allow direct use of OP_ADD and OP_SUB for addition and subtraction operations. While there is no OP_MUL for multiplication, it can be implemented using existing opcodes through the double-and-add algorithm.

However, Bitcoin currently does not support the OP_CAT opcode. To implement a hash function-based ZK Verifier on Bitcoin, the following strategies could be considered:

- Implement a Bitcoin-friendly hash function using existing opcodes. The script size for the currently implemented Blake3 hash function is still too large. Exploring the structural properties of small finite fields, such as the bit shift for multiplication by a power of 2 feature in M31, could enable the implementation of hash functions like Poseidon.

There are several ways to optimize the ZK-Proof Verifier on Bitcoin:

- Optimize the existing Winternitz signature to reduce the state transfer overhead between scripts.
- Split the Bitcoin script into multiple transactions to execute the ZK Verifier algorithm.
- Use an OP mechanism that adjudicates on-chain only in case of disputes. The OP solution ensures that as long as there is one honest participant, the correct result can be guaranteed with acceptable costs.
- Adopt a ZK Fraud proof scheme that generates ZK-Proof only in case of disputes. This On-Demand mode can reduce off-chain Prover costs.
- Different ZK-Proof systems can be combined, balancing the off-chain Prover costs and on-chain Verifier costs.

Implementing a ZK-Proof Verifier on Bitcoin has profound implications for the Bitcoin ecosystem. It not only helps to achieve Bitcoin transaction introspection but also enables the creation of a ZK aggregation layer. This layer can aggregate proofs from multiple Bitcoin ecosystem projects into a single on-chain verification, significantly alleviating the congestion issue of the Bitcoin network.

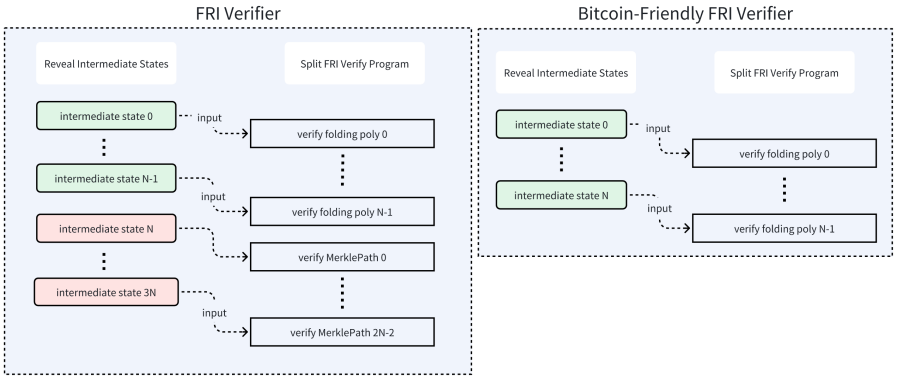
3.2.4 Bitcoin Friendly FRI

The creative work we have done at Zero-Knowledge Proof Verification on Bitcoin is to provide an FRI protocol that can be verified on Bitcoin at very low cost, called Bitcoin Friendly FRI.

In the above Challenge Game, the Prover reveals all states' preimages on the chain. The Verifier, on the other hand, obtains the Prover's staked amount by proving that the states revealed by the Prover would cause the corresponding gates to execute incorrectly.

However, the issue here is that the Prover needs to reveal all states on the chain, which can be costly if the computation complexity is high. Even if we pack each leaf script with 400KB, there are still many intermediate states. To verify STARK, we need to verify an excessive number of Merkel paths, which would make our verification program large and thus cause too many intermediate states.

During the upgrade process of Bitcoin's Taproot, the concept of Taptree was introduced. We can replace MerkleTree with Taptree by committing all



	BF FRI Verifier	FRI Verifier
Prover Reveal Transaction Size	N x 2 x field_size	N x 2 x field_size +MerklePath_size x N x 2
Intermediates State Number	N	3N

Fig. 10. Bitcoin-friendly FRI

evaluations corresponding to the committed polynomial in the Leaf Script of Taptree. In this way, we can avoid simulating the verification process of Merkel paths on the chain using Bitcoin Script. At the same time, Prover can reduce the cost of revealing intermediate states on the chain. Our ultimate goal is to continue to reduce the cost of folding polynomials on top of this so that validity proofs can be submitted to Bitcoin.

In the following diagram, for the FRI Verifier, the number of intermediate states prover needed to reveal is $3N$. Due to the Bitcoin Friendly FRI Verifier avoiding verifying MerkleTree Path on chain, it only needs to reveal N intermediate states. If you are interested in Bitcoin-Friendly FRI, you can get more information here.

3.2.5 Scaling security: Multi-proofs challenge game

Implementing zkproof verifiers based on Bitcoin scripts is an extremely challenging task, achieving bug-free status is nearly impossible. Inspired by the principle of diverse clients in the blockchain realm, Bitlayer plans to utilize Bitcoin and BitVM technologies to establish a variety of zkproof verification challenge systems, including ZK-SNARKs (such as Groth16) and the Bitcoin-friendly ZK-STARKs mentioned earlier.

The diagram above illustrates the operational principle of Bitlayer's multi-proofs challenge game system. Bitlayer Prover will simultaneously create multiple fund transactions on Bitcoin, targeting different types of ZK-Proof verifier challenge programs. Any verifier who successfully challenges one of these proof

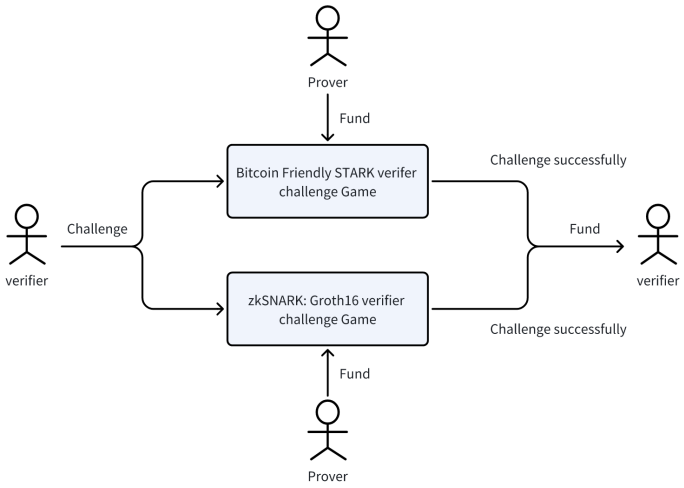


Fig. 11. Multi-proofs challenge game

verifiers will be rewarded. This effectively reduces the risk associated with a single proof verifier challenge system and scales Bitlayer's security.

3.3 Layered Virtual Machine

Layered virtual machine (LVM) technology represents a significant advancement in terms of smart contract execution. This innovative approach enables the support of various frontend smart contract types, such as EVM/CairoVM, and backend zero-knowledge proof verifiers, including ZK-STARKs or ZK-SNARKs(groth16, plonk, etc.) while maintaining robust security measures and enhancing computational flexibility.

Supporting multiple frontend smart contract types can easily attract developers from various blockchain ecosystems. For instance, EVM is a widely used smart contract platform with its own bytecode and execution environment. By integrating support for EVM within the LVM framework, developers can leverage existing EVM-based smart contracts seamlessly while also benefiting from enhanced execution capabilities provided by layered architecture.

On the backend side, supporting various zero-knowledge proof generators provides more optimization possibilities for the disputing process without compromising existing functionalities. Especially in the future, when more succinct and efficient zero-knowledge proof systems emerge, Bitlayer can quickly adopt and switch, which is essential for keeping pace with evolving blockchain standards, emerging technologies, and changing user requirements.

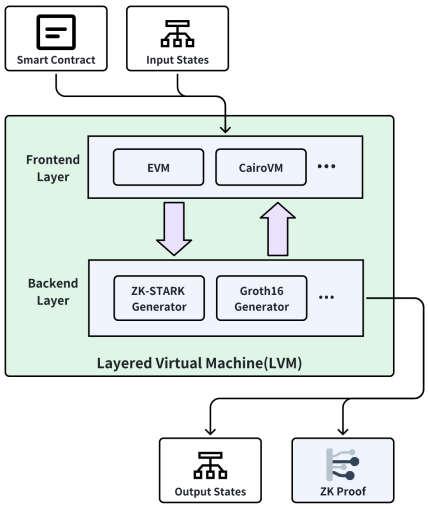


Fig. 12. Layered virtual machine

4 Assets Bridge

Bitlayer introduces a dual-channel Two-Way Peg asset bridge, comprising the OP-DLC (DLC with optimistic mechanism) bridge and BitVM bridge, which operate concurrently. This architecture provides tailored security options to meet diverse user preferences. The OP-DLC Bridge is a channel that enables users to lock BTC in a trustless and self-controlled manner. By incorporating our novel challenge protocol, OP-DLC effectively addresses the collusion issue associated with Oracles in the original DLC protocol. The BitVM channel, on the other hand, facilitates a minimally trusted custodian scheme, requiring only a 1 of N security level. For those seeking robust security and self-controlled assets, the OP-DLC channel is available, while the BitVM channel caters to users requiring greater flexibility and speed. Furthermore, the dual-channel design enhances scalability, ensuring efficient performance even under high demand.

4.1 Threat Model

We operate under the premise that the Bitcoin blockchain maintains robust security, effectively immunizing it against vulnerabilities like double-spending attacks. Nonetheless, there exists a potential risk within the BitVM Federation nodes, tasked with managing the dual-channel bridge, where they might be susceptible to compromise. Our security model acknowledges the possibility that an adversary could gain control over up to $n-1$ nodes, implying that the integrity of the system can be upheld as long as a single host node remains uncompromised and operational.

For the uncompromised node, the adversary cannot read or tamper with the data and code running on the node. The uncompromised node serves as a verifier in the BitVM model, initiating a challenge game if the BitVM bridge operator or the OP-DLC oracle nodes act maliciously. In the challenge game, the malicious node must provide fraud proof to win; otherwise, the honest verifier prevails and claims the prover's deposit.

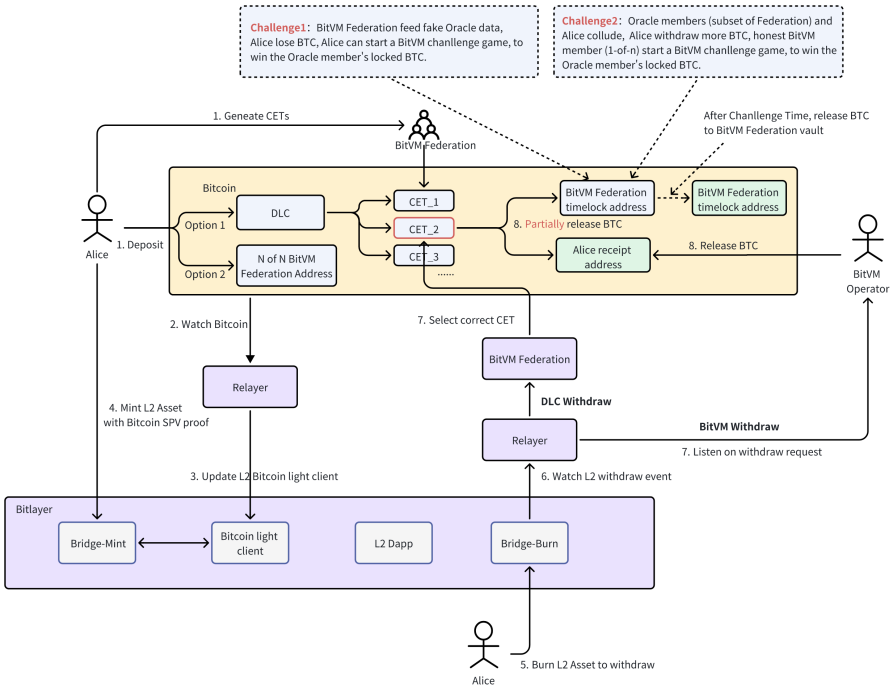


Fig. 13. Bitlayer Bridge

4.2 System Architecture

The Bitlayer assets bridge provides user-controlled, decentralized custody, and high-liquidity Bitcoin Layer 2 cross-chain solution based on BitVM+DLC technology. Bitlayer offers a dual-channel two-way peg bridge that not only meets the self-controlled asset needs of Layer 1 users for BTC deposit and withdrawal but also satisfies the smooth withdrawal requirements of native Layer 2 users. The core components of the assets bridge consist of BitVM Federation nodes, DLC components, Layer 2 smart contracts, and Relayers:

1. **BitVM Federation:** The nodes within the BitVM Federation act as a verification network to ensure the secure execution of Layer 2 transactions and

the stable operation of the Bitlayer Bridge. When Layer 2 begins, validated organizations can join the federation by depositing a specific amount of BTC. As the Layer 2 network progresses, the federation dynamically adjusts and increases its membership to boost security and decentralization. Within the cross-chain bridge, the BitVM Federation collectively manages decentralized asset custody for the BitVM bridge channel, attaining a 1 of N security level, meaning only one honest node is needed for network integrity. Moreover, the BitVM Federation functions as an oracle network for the OP-DCL bridge channel, requiring only some members to agree (t of N) to produce a legitimate oracle signature.

2. **DLC Components:** Using DLC for deposits and withdrawals ensures users' autonomous control over their assets but introduces restrictions on the BTC amount for deposits and withdrawals. Because DLC requires predefined CETs to determine the withdrawal amount, supporting fine-grained CETs is necessary to meet user-friendly withdrawal requirements. The first function of the DLC component is to facilitate the creation of funding transactions, where the assets are initially output to a 2 of 2 multi-sig output, with the parties involved being the user and the BitVM Federation (N of N) address. The second function is the CET manager, which pre-creates DLCs supporting multiple future withdrawal requirements, thus realizing a user-friendly cross-chain solution.

3. **Layer 2 Smart Contracts:** The Bridge and Light Client are two core smart contracts on Layer 2 that implement the trustless Bridge. The Bridge smart contract manages the issuance and destruction of BTC assets on Layer 2. The Light Client contract maintains Bitcoin block header information on Layer 2, and Bitlayer uses ZKP-based Bitcoin state proofs to update and maintain the block header information. The Light Client contract also provides a Verify function to validate Bitcoin transactions, by submitting a Simplified Payment Verification (SPV) proof of the transaction to the Light Client contract to verify the legitimacy of the Bitcoin transaction. The Bridge contract calls the Light Client's Verify function to validate the legitimacy of users' locking transactions on Bitcoin, ensuring that all BTC assets on Layer 2 are issued in a Trustless manner.

4. **Relayers:** The Relayer plays a critical, trustless role in the Bitlayer Asset Bridge, primarily tasked with monitoring both Layer 1 and Layer 2 blockchains and updating the state of Light Client data on the Layer 2 blockchain. When the Bitcoin network commits a new block, the relayer submits a state update transaction for the Light Client, accompanied by a zero-knowledge proof. Whenever there is a bridge transaction, the relayer forwards it to a smart contract (Peg-In) or a BitVM Federation node (Peg-Out) for further processing. The inclusion of this permissionless relayer system ensures the continuous operation of the Asset Bridge; the bridge remains functional as long as at least one relayer is operational.

4.3 Bridge Protocol

We present our Bridge protocol, which encompasses a specific lifecycle comprising Peg-In and Peg-Out processes, as illustrated in the figure above.

4.3.1 Peg-In

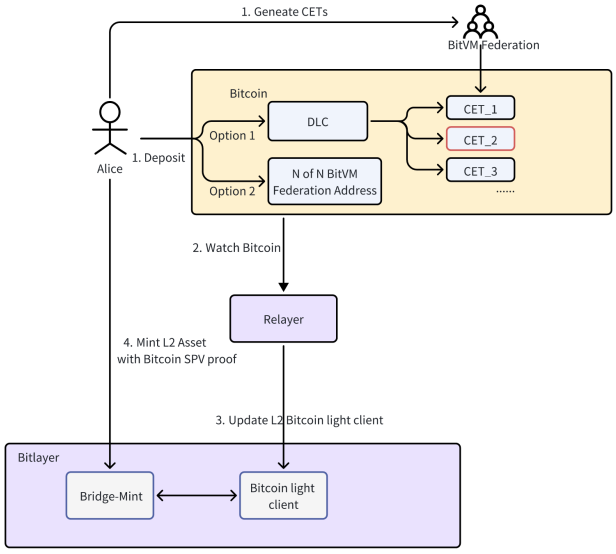


Fig. 14. Bitlayer Bridge Peg-In

The user Alice initiates a deposit request and locks a certain amount of BTC on the Bitcoin network. Alice has two deposit options: Option 1: Deposit via OP-DLC Channel - Alice locks BTC into a DLC contract and negotiates with the BitVM Federation to create CETs for various withdrawal needs (the smallest unit is 0.1 BTC). Option 2: Deposit via BitVM Bridge Channel - Alice directly locks BTC into an N of N multi-sig address of the BitVM Federation, without needing to consider future withdrawal requirements. Future withdrawals can also be made directly through the BitVM Bridge (any amount, faster speed). For both deposit channels, user Alice will provide a receipt address (such as an EVM address) in the Bitcoin script to serve as the recipient on Layer 2. This process is depicted in Step 1.

The Relayer node continually watches the Bitcoin network, and is responsible for submitting the latest block information from Bitcoin to the Layer 2 Light Client smart contract. This process is depicted in Step 2 and 3.

Once the user's deposit locking transaction has reached the required number of confirmation blocks (usually 7 blocks), any party can submit a Mint transaction to the Layer 2 bridge contract, accompanied by an SPV proof to validate the transaction. The bridge contract will assess the Mint transaction's legitimacy. Upon confirming that the locking transaction is valid, it will execute the Mint operation, thereby minting the equivalent BTC to Alice's designated receipt address on Layer 2. To safeguard against double-spending attacks, the bridge contract maintains a record of the state of each Peg-In transaction. This process is depicted in Step 4.

4.3.2 Peg-Out

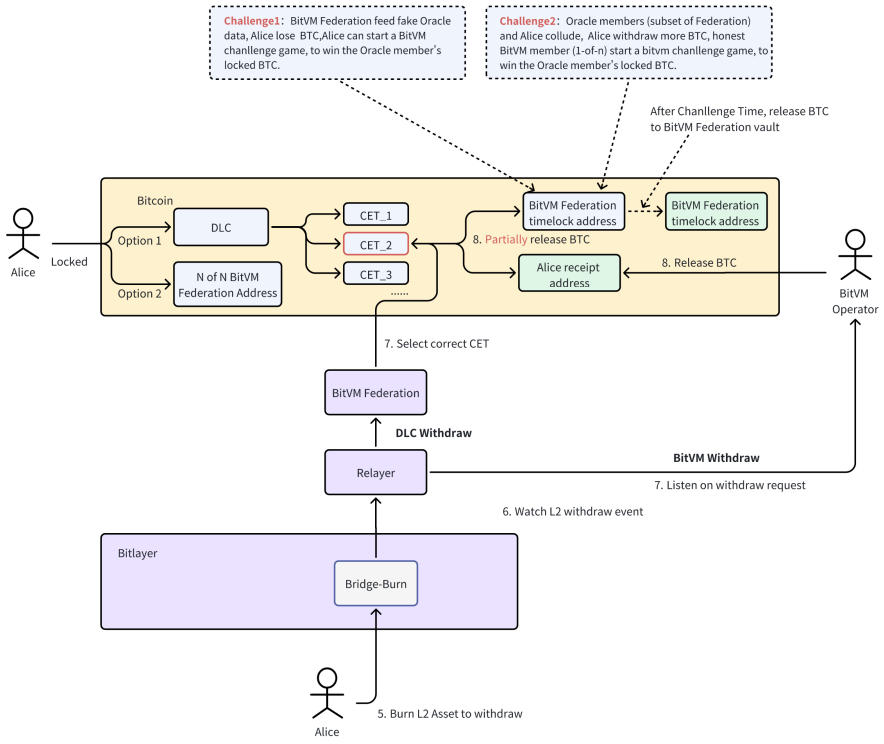


Fig. 15. Bitlayer Bridge Peg-Out

User Alice submits a withdrawal request and burns a specified amount of BTC on the Bitlayer network. She then has two withdrawal options: Option 1 allows her to withdraw directly via the BitVM Bridge Channel, while Option 2 employs a joint withdrawal mechanism using OP-DLC+BitVM Bridge Channel, with OP-DLC handling the main withdrawal and BitVM used for smaller

amounts or change. Specifically, if Alice initially deposited BTC through the OP-DLC channel, she has the flexibility to choose either withdrawal option. However, if her initial deposit was via the BitVM Bridge Channel, or if she is an original Layer 2 user without a Peg-In transaction, her withdrawal is limited to the BitVM Bridge Channel. This step is illustrated in Step 5.

The Relayer node will consistently observe the Bitlayer network for any withdrawal activities. Once it identifies a withdrawal event, it will assess the chosen method of withdrawal before advancing to the next steps. Subsequently, it relays the withdrawal request to the appropriate BitVM Federation nodes for further action. This step is illustrated in Step 6.

If Alice chooses a direct withdrawal through the BitVM Bridge Channel, a BitVM Bridge operator will first cover the withdrawal amount for Alice, allowing her to receive the equivalent BTC on the Layer 1 network. After providing the funds, the BitVM Bridge operator can periodically reclaim the advanced BTC from the BitVM Federation's multi-sig (n of n) address. For security reasons, this claim request must successfully pass the BitVM challenge process. If the BitVM Bridge operator acts maliciously, they will fail the challenge process and lose their deposit. Another way is if the user opts for a collaborative withdrawal through the OP-DLC+BitVM Bridge Channel, they must have a DLC contract pre-locked on Bitcoin. The withdrawal amount should be equal to or greater than a predetermined CET output. The withdrawal will consist of both the OP-DLC portion and the BitVM change portion, with the latter being handled just like a standard BitVM withdrawal. This is illustrated in Step 7 and 8.

Details of the OP-DLC Peg-Out Channel: After receiving an OP-DLC withdrawal request, BitVM Federation nodes will select a matching CET output from the CETs manager, and a quorum of t BitVM Federation nodes will collaboratively sign to generate the effective DLC signature for that CET. Upon receiving the broadcasted signature data, Alice can immediately receive the corresponding BTC from one of the CET outputs. The BTC directed to the BitVM Federation address is first entered into a timelock script, allowing Alice to challenge the BitVM Federation members' oracle signature data. The BitVM Federation address will receive the corresponding BTC only after the challenge period ends. The two challenge games are as follows:

- Challenge 1: If the BitVM Federation members involved in the oracle sign incorrect data, causing Alice to receive less BTC than she's entitled to, Alice has the right to initiate a BitVM challenge. Through this challenge, she can claim the BTC that was locked by the Oracle members involved.
- Challenge 2: If the BitVM Federation members participating as oracles conspire with Alice, resulting in her receiving more BTC than she's entitled to, this action undermines the interests of the entire BitVM Federation. In such cases, the honest members of the BitVM Federation have the authority to start a challenge against the colluding members, aiming to claim the BTC they had locked.

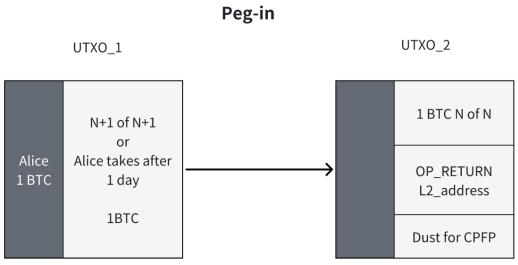


Fig. 16. UTXO_1 and UTXO_2 during Peg-In

4.3.3 Forced Withdrawal

In Bitlayer, BitVM Federation is also the Oracle and the counterparty in the OP-DLC bridge. Both the locking transaction in the BitVM bridge and the funding transaction in the OP-DLC bridge will be split into two transactions for anti-censorship.

To make a Peg-In, user Alice locks 1 BTC into a UTXO_1 that is only spendable by either N+1 of N+1 multi-sig, or, Alice after 1 day, where there are N-1 verifiers, 1 bridge operator and 1 user. Then Alice and BitVM Federation co-sign to spend UTXO_1 and get UTXO_2, including an Layer 2_address to receive wrapped BTC in Layer 2.

If the BitVM Federation censors Alice's Peg-In, without co-signing to spend UTXO_1, then after 1 day, Alice can take back her funds by herself. If the BitVM Federation collaborates to spend UTXO_1 and get UTXO_2, then Alice can use the Bitcoin SPV proof to mint the same amount of locking amount in Layer 1 to the specific Layer 2_address as shown in UTXO_2.

When something is wrong in Layer 2, Alice can use the DA and open-source software to reconstruct the Layer 2 state. When Alice wants to Peg-Out through OP-DLC, she needs t signatures out of n oracles to sign the right CET, or collaborates with the counterparty to close the OP-DLC channel. When Alice wants to Peg-Out through BitVM bridge, she needs 1 honest verifier to help her.

4.4 Security Analysis

In this section, we demonstrate how the Bitlayer Bridge maintains the properties of safety and liveness within the context of the threat model presented in Section 4.1.

4.4.1 Security Analysis

For the Peg-In bridge, both OP-DLC and BitVM bridge channels secure safety by relying on the integrity of the Light client, which employs zero-knowledge proofs. The system ensures that minting of the equivalent BTC on the Bitlayer

network occurs only if there is a valid locking transaction on the Bitcoin network, verified by the Light client contract.

For the Peg-Out bridge, we conduct separate analyses for the OP-DLC and BitVM bridge channels as follows:

OP-DLC Channel

In Bitlayer's OP-DLC bridge channel, both the Oracle and Bob roles are consolidated under a single entity called the BitVM Federation. This federation manages a multi-signature liquidity pool and concurrently acts as the Oracle.

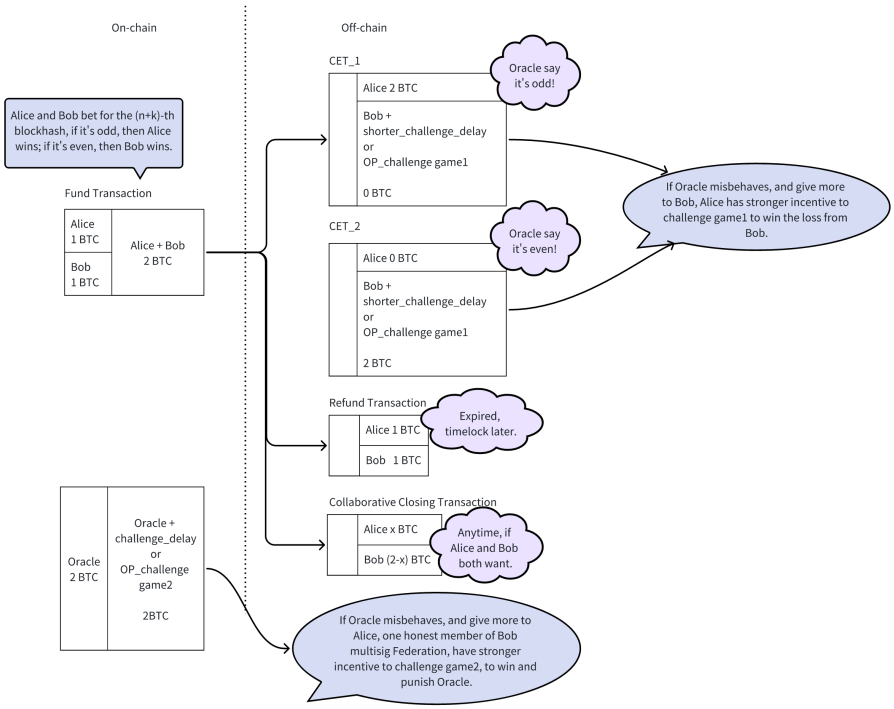


Fig. 17. OP-DLC Challenge games

Under the threat model in Section 4.1, there is at least one honest node in the BitVM Federation. We demonstrate that the OP-DLC channel Peg-Out achieves safety by introducing two separate Challenge games:

– Game 1: If the Oracle misbehaves and allocates more to Bob, Alice has a stronger incentive to challenge Game 1 to recover her losses from Bob. This mechanism ensures that Oracle operates correctly, thereby maintaining safety through an incentive activity.

– Game 2: If the Oracle misbehaves and allocates more to Alice, an honest member of Bob's multi-signature Federation is highly motivated to challenge Game 2, to win and penalize the Oracle. This setup ensures that both the Oracle and Alice act appropriately, safeguarding safety through an incentive activity.

All the Challenge games are facilitated through our novel Challenge Protocol, as detailed in Section 3.2, ensuring an efficient and secure process.

BitVM Bridge Channel

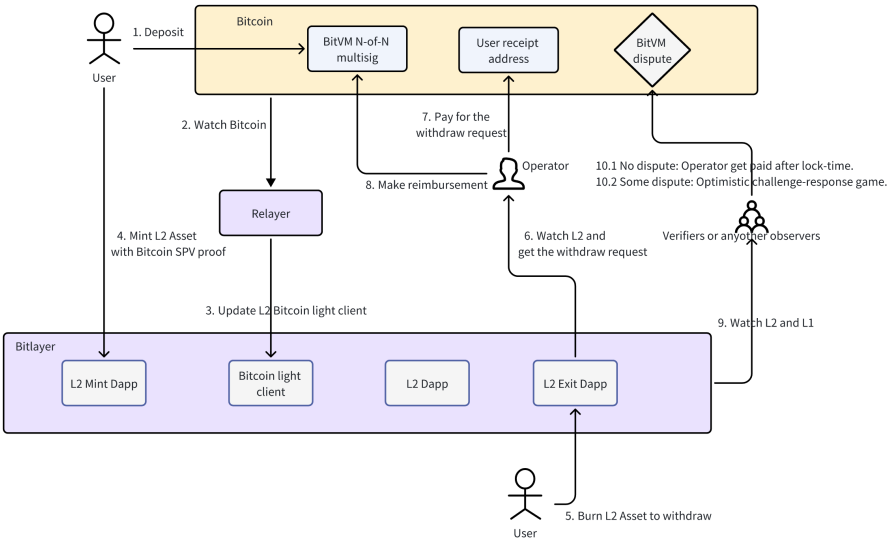


Fig. 18. BitVM Bridge channel

In the context of the BitVM bridge, the threat model assumes there is at least one honest node within the BitVM Federation. To demonstrate how the system ensures safety, we examine two potential scenarios:

– Case 1: The bridge operator is the honest node. This operator adheres to the correct protocol by advancing the specified BTC amount to the user on the Bitcoin network and subsequently reclaiming the same amount from the BitVM Federation's multi-sign address. The system functions correctly, thereby ensuring safety.
– Case 2: The bridge operator is malicious. In this scenario, a dispute arises when the operator attempts to reclaim more than the advanced BTC amount. An honest node identifies the discrepancy and initiates a BitVM challenge game to take the locked BTC of the malicious node and impose a penalty. Through this incentive-driven mechanism, the system achieves safety.

4.4.2 Liveness Analysis

For the Peg-In bridge, both OP-DLC and BitVM bridge channels maintain liveness, provided at least one relay operates effectively, as mentioned in the previous section. According to the threat model, there's at least one honest node within the BitVM Federation. Given that the relayer is open-source and permissionless, the honest node can perform relayer services to ensure the system's liveness.

For the Peg-Out bridge, the system attains two levels of liveness: strong liveness and weak liveness, applicable to the two distinct channels. The Peg-Out process of the BitVM bridge achieves strong liveness, requiring only one honest node to be operational within the BitVM Federation. Similar to the Peg-In process, this honest node can function as both the relayer and the bridge operator, ensuring that bridge transactions are finalized and liveness is achieved.

On the other hand, the Peg-Out process of the OP-DLC bridge attains weak liveness, which necessitates at least $t(t \leq n)$ honest nodes to be active within the BitVM Federation. Since the Oracle network requires t nodes to sign a valid signature, the system only achieves liveness when there are enough t nodes operational.

5 Conclusion

By the technological design described above, Bitlayer's comprehensive approach not only addresses scalability challenges but also prioritizes security and flexibility, positioning itself as a leading solution in the evolving Bitcoin Layer 2 landscape. Its innovations hold promise for unlocking new possibilities in decentralized finance and blockchain applications, driving continued growth and adoption in the crypto industry.

In the future, Bitlayer will continue to optimize and enhance the transaction verification mechanism on Bitcoin, such as introducing zero-knowledge proofs to improve the efficiency of the Challenge-Response process. Additionally, continuous research and development will be conducted on the Layered Virtual Machine (LVM) to support more smart contract development languages. Moreover, Bitlayer will also explore new cross-chain solutions to enhance the security of user assets on the Bitlayer platform.

References

1. Robin Linus. BitVM: Compute Anything on Bitcoin, 2023
2. Thaddeus Dryja. Discreet Log Contracts, 2017
3. Ethereum Research. Optimistic rollups, 2022.
4. Optimism Foundation. Optimism Docs.
5. Lee Bousfield, Rachel Bousfield, Chris Buckland, et al. Arbitrum Nitro: A Second-Generation Optimistic Rollup, 2022
6. Masip-Ardevol, H., Guzmán-Albiol, M., Baylina-Melé, J., & Muñoz-Tapia, J. L. (Year). eSTARK: Extending STARKs with Arguments. Universitat Politècnica de Catalunya; Polygon zkEVM.

7. Haböck, U. (Year). A summary on the FRI low degree test. Polygon Labs; Orbis Labs; Horizen Labs, Inc.
8. Polyhedra. How to verify ZK proofs on Bitcoin?, 2024
9. BitVM org. SNARK Verifier in Bitcoin Script, 2024
10. BitVM org. BitVM 2: Permissionless Verification on Bitcoin, 2024
11. Citrea Team.Unveiling Clementine - Citrea's BitVM Based Trust-Minimized Two-Way Peg Program, 2024
12. Robin Linus. BitVM: Advancing Bitcoin Contracts, 2024
13. Specification for Discreet Log Contracts.
14. Bitlayer Labs, BitVM And Its Optimization Considerations, 2024
15. Bitlayer Labs, Bitlayer Core Technology: DLC and Its Optimization Considerations, 2024