XPower

A Proof-of-Work Token System with NFT Staking and Time-Weighted Rewards

Karun The Rich*

Abstract

We present XPower, a novel proof-of-work (PoW) token system deployed on the Avalanche blockchain that combines cryptographic mining with NFT-based staking and time-weighted reward mechanisms. Unlike traditional PoW cryptocurrencies that require dedicated mining infrastructure, XPower enables browser-based mining through nonce discovery using the Keccak-256 hash function. The protocol introduces a dual-token economy: XPOW (the base mining token) and APOW (an aged store-of-value derivative), complemented by ERC-1155 NFTs that serve as staking instruments. We describe the mathematical foundations of the difficulty-to-reward mapping, the polynomial-based annual percentage rate (APR) system, and the integrator mechanism for computing duration-weighted means. Our implementation includes anti-gaming protections, migration support for protocol upgrades, and dynamic rate adjustment based on staking distribution. The system achieves fair token distribution through computational work while providing long-term value accrual through the staking and aging mechanisms.

1 Introduction

The distribution of cryptocurrency tokens remains a fundamental challenge in decentralized systems. Initial Coin Offerings (ICOs) and token airdrops suffer from centralization concerns and Sybil attacks, while traditional proof-of-work mining requires significant capital investment in specialized hardware [1]. XPower addresses these limitations by implementing a browser-accessible proof-of-work system where tokens are minted proportionally to computational difficulty achieved.

The XPower protocol consists of six interconnected smart contracts:

1. **XPower (XPOW)**: The base ERC-20 token minted through proof-of-work
2. **XPowerNft**: ERC-1155 NFTs representing deposited XPOW tokens
3. **APowerNft**: ERC-1155 NFTs tracking staking duration (staking receipts)
4. **APower (APOW)**: A rate-limited store-of-value token
5. **MoeTreasury**: The rewards distribution and APR/APB management contract
6. **NftTreasury**: The staking/unstaking management contract

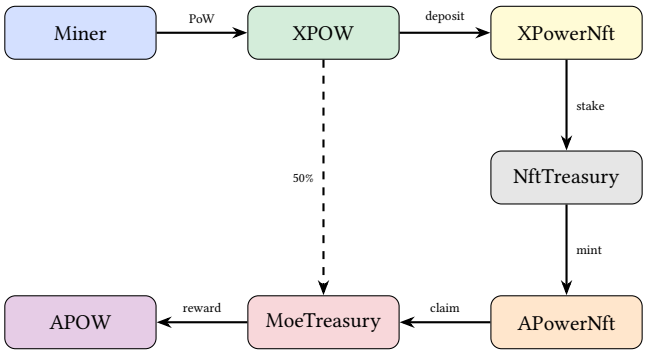

Figure 1: XPower protocol architecture: Token flow from mining through staking to rewards. NftTreasury manages stake/unstake operations while MoeTreasury handles reward distribution.

This paper is organized as follows: Section 2 discusses related work. Section 3 describes the proof-of-work mechanism. Section 4 details the NFT staking system. Section 5 presents the reward calculation methodology. Section 6 covers the upgrade mechanism. Section 7 discusses security considerations. Section 8 concludes.

2 Related Work

2.1 Proof-of-Work Cryptocurrencies

Bitcoin [1] pioneered the use of proof-of-work for distributed consensus, using SHA-256 double hashing with adjustable difficulty. Ethereum [2] employed Ethash, a memory-hard PoW algorithm designed to resist ASIC mining, before transitioning to proof-of-stake. Monero uses RandomX [3], optimized for general-purpose CPUs.

XPower differs from these systems in that PoW is used solely for token distribution, not consensus. The blockchain

*https://www.xpowermine.com

consensus is delegated to Avalanche's proof-of-stake mechanism, while mining determines individual token rewards.

2.2 Token Staking Mechanisms

DeFi protocols have developed various staking mechanisms. Compound [4] introduced yield-bearing tokens (cTokens) that accrue interest continuously. Lido [5] created liquid staking derivatives for proof-of-stake networks. Curve [6] implemented vote-escrowed tokens (veCRV) with time-locked staking.

XPower's approach combines NFT representation with time-weighted reward calculation, allowing transferable staking positions while tracking individual stake duration.

2.3 ERC-1155 Token Standard

The ERC-1155 multi-token standard [7] enables efficient batch operations and unified handling of fungible and non-fungible tokens. XPower leverages this standard for staking NFTs, encoding both the issuance year and denomination level in the token identifier.

3 Proof-of-Work Mining

3.1 Hash Function and Nonce Discovery

XPower mining requires discovering a nonce n such that the hash H of the mining parameters contains a specified number of leading zero nibbles. The hash is computed as:

$$H = \text{Keccak256}(A \oplus B \parallel b \parallel n) \tag{1}$$

where:

- A is the contract address (20 bytes)
- B is the beneficiary address (20 bytes)
- b is a recent block hash (32 bytes)
- n is the nonce (variable length)
- $\oplus$ denotes XOR operation
- $\parallel$ denotes concatenation

The XOR of contract and beneficiary addresses prevents nonce reuse across different recipients while maintaining a fixed-size prefix.

3.2 Difficulty and Token Amount

The difficulty level z is defined as the number of leading zero nibbles (half-bytes) in the hash result. Given a 256-bit hash, z ranges from 0 to 64. The token amount minted is:

$$\text{amount}(z) = (2^z - 1) \times 10^{18} \tag{2}$$

This exponential relationship ensures that higher difficulties yield proportionally more tokens:

Table 1: Difficulty levels and token rewards

Zeros (z)	Amount (XPOW)	Probability
1	1	$1/16$
2	3	$1/256$
3	7	$1/4096$
4	15	$1/65536$
$\vdots$	$\vdots$	$\vdots$
z	$2^z - 1$	$1/16^z$

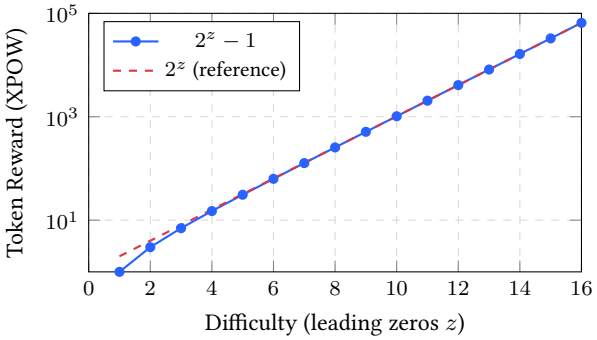

Figure 2: Exponential growth of token rewards with mining difficulty. Each additional leading zero doubles the reward.

3.3 Block Hash Intervals

To prevent pre-computation attacks, nonces must reference a block hash from the current hour interval:

$$\text{interval} = \left\lfloor \frac{t}{3600} \right\rfloor \tag{3}$$

where t is the current Unix timestamp. Block hashes are cached per interval via the `init()` function, which must be called to establish the reference hash.

The validity check ensures a block hash is recent:

$$\text{timestamp}[\text{blockHash}] \geq \text{interval} \times 3600 \tag{4}$$

Nonces submitted with expired block hashes are rejected. This mechanism was identified as critical during the security audit and correctly implemented to prevent inflation attacks [11].

3.4 Duplicate Prevention

Each (nonce-hash, block-hash) pair can only be used once. The contract maintains a mapping of used pair indices:

$$\text{pairIndex} = H_{\text{nonce}} \oplus b \tag{5}$$

This ensures that even if two miners discover the same nonce for different block hashes, both submissions are valid.

3.5 Treasury Allocation

Upon successful mining, tokens are minted to both the beneficiary and the contract owner (treasury) in equal amounts:

$$\text{totalMint}(z) = 2 \times \text{amount}(z) \tag{6}$$

This 50% allocation to the treasury funds the staking reward system described in Section 5.

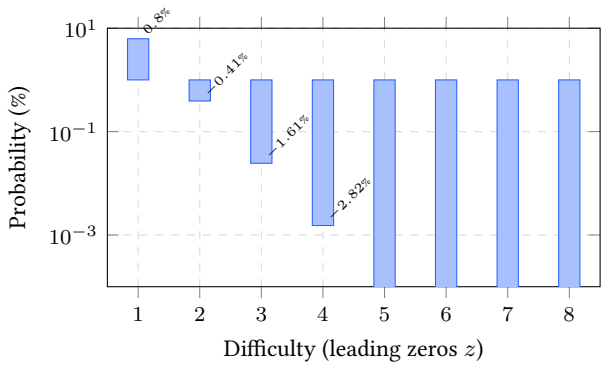

Figure 3: Mining success probability by difficulty level. Probability decreases by factor of 16 with each additional zero.

4 NFT Staking System

4.1 XPowerNft: Staking Positions

Users convert XPOW tokens to XPowerNft positions by depositing tokens. Each NFT has a denomination level ℓ (where $\ell \mod 3 = 0$, i.e., $\ell \in \{0, 3, 6, \ldots, 99\}$) and an issuance year y. The NFT identifier encodes both:

$$\text{nftId} = 100 \times y + \ell \tag{7}$$

The denomination of level ℓ is:

$$\text{denomination}(\ell) = 10^{\ell} \tag{8}$$

Table 2: NFT levels and denominations

Level (ℓ)	Denomination	Name
0	10^0	Unit
3	10^3	Kilo
6	10^6	Mega
9	10^9	Giga
⋮	⋮	⋮

4.2 NFT Upgrading

Users can upgrade 1,000 NFTs of level ℓ to 1 NFT of level $\ell + 3$:

$$1000 \times \text{NFT}_{\ell} \rightarrow 1 \times \text{NFT}_{\ell+3} \tag{9}$$

This maintains the total token value while consolidating positions.

4.3 Redemption Maturity

NFTs can be redeemed back to XPOW tokens after a maturity period. An NFT with level ℓ and year y becomes redeemable when:

$$y + 2^{\lfloor \ell/3 \rfloor} - 1 \leq y_{\text{current}} \tag{10}$$

Higher-level NFTs have longer lock periods, incentivizing long-term commitment.

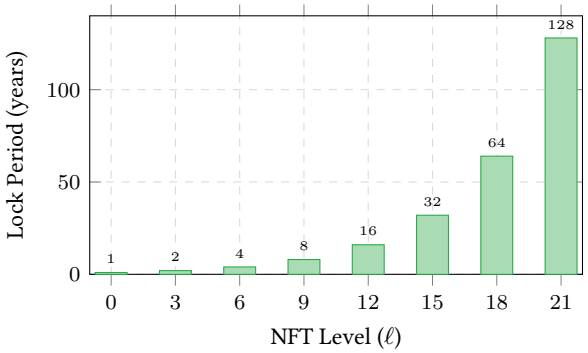

Figure 4: NFT maturity periods by level. Lock duration doubles every 3 levels following $2^{\lfloor \ell/3 \rfloor}$ years.

4.4 APowerNft: Staking Receipts

When XPowerNfts are staked in the NftTreasury, users receive APowerNfts as receipts. These track:

1. The staked amount per NFT identifier
2. The weighted staking timestamp (age accumulator)
3. The total shares per level for rate adjustment

The age accumulator A for an account and NFT is updated on mint/burn:

$$A_{\text{mint}} = A + \text{amount} \times t \tag{11}$$

$$A_{\text{burn}} = A - A \times \frac{\text{amount}}{\text{balance}} \tag{12}$$

The effective staking age is computed as:

$$\text{age} = \text{balance} \times t_{\text{current}} - A \tag{13}$$

This proportional adjustment ensures that partial unstaking preserves the average staking duration per remaining token (see Appendix for proof). When APowerNfts are transferred, the `refreshClaimed` function proportionally rescales accumulated rewards to maintain fairness.

5 Reward Calculation

5.1 APower: Store-of-Value Token

APower (APOW) is the reward token minted for stakers. It wraps XPOW tokens at a rate-limited pace, targeting one

token per minute on average. The minting formula incorporates a moving mean to smooth supply:

$$m_t = \frac{m_{t-1} \times t_{\text{prev}}}{t} + \frac{\sqrt{c}}{t} \tag{14}$$

$$\text{mintable} = \frac{\sqrt{c} \times 10^{18}}{60 \times m_t} \tag{15}$$

where c is the claim amount and t is time since contract deployment. The square root transformation reduces inflation from large claims.

5.2 Reward Formula

The reward for a staked NFT position is:

$$R = \frac{(\text{APR} + \text{APB}) \times \text{age} \times 10^{18} \times D}{10^6 \times C} \tag{16}$$

where:

- APR is the annualized percentage rate (per level)
- APB is the annualized percentage bonus (per year)
- age is the staking duration in seconds
- D is the NFT denomination
- C is a century in seconds (3,155,760,000)

5.3 APR Polynomial Calculation

The APR is computed using a polynomial evaluation with fractional exponentiation:

$$\text{APR}(\ell) = \frac{\left(\text{add} + \frac{\ell \times \text{mul}}{\text{div}}\right)^{\text{exp}/256} \times 10^6}{\text{base}^{\text{exp}/256}} \tag{17}$$

The default parameters are:

- add $= 0$
- div $= 3$
- mul $= 3,375,000$ (representing 3.375%)
- exp $= 256$ (unity exponent)

This yields a linear APR growth with NFT level: $\text{APR}(\ell) = 1.125\ell\%$, so each ternary level step (e.g., from level 3 to level 6) adds 3.375% APR.

5.4 APB Polynomial Calculation

The annual percentage bonus rewards older NFT vintages:

$$\text{APB}(y) = \frac{\left(\text{add} + \frac{(y_{\text{current}} - y_{\text{nft}}) \times \text{mul}}{\text{div}}\right)^{\text{exp}/256} \times 10^6}{\text{base}^{\text{exp}/256}} \tag{18}$$

Default APB is 0.1% per year of NFT age (100,000 basis points, div=1, exp=256).

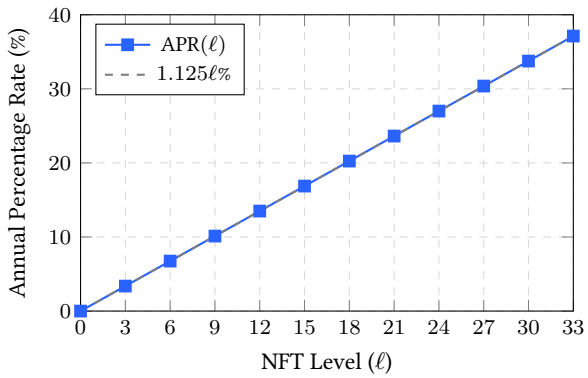

Figure 5: Annual Percentage Rate (APR) by NFT level. APR increases linearly at 1.125% per level, so higher denomination NFTs earn proportionally higher rewards.

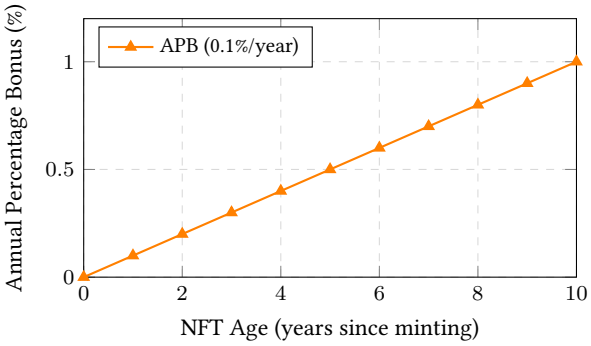

Figure 6: Annual Percentage Bonus (APB) increases linearly with NFT vintage age, rewarding long-term holders.

5.5 Dynamic Rate Adjustment

The MoeTreasury implements dynamic APR adjustment based on staking distribution. The shares accumulator tracks:

$$\text{shares}[\ell/3] = \sum_{\text{stakes}} \text{amount} \times 10^{\ell} \tag{19}$$

The scalar adjustment for each level is:

$$\text{scalar}(\ell) = \frac{\text{mul} \times \sum \text{shares}}{\text{bins} \times \text{shares}[\ell/3]} \tag{20}$$

where bins is the number of active staking levels. This redistributes rewards toward less-staked levels, incentivizing diversification.

5.6 Integrator Mechanism

Rate changes are smoothed using a time-weighted integrator. Each rate update appends (t, v) to the history, computing cumulative area:

$$\text{area}_n = \text{area}_{n-1} + v_n \times (t_n - t_{n-1}) \tag{21}$$

The duration-weighted mean is:

$$\bar{v} = \frac{\text{area}_n + v_{\text{current}} \times (t_{\text{current}} - t_n)}{t_{\text{current}} - t_0} \tag{22}$$

This prevents rate manipulation by averaging over the full staking period.

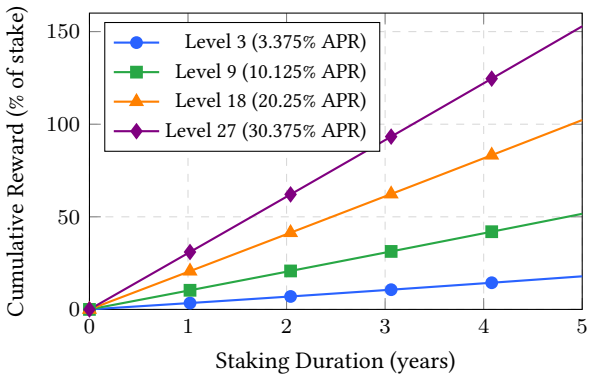

Figure 7: Cumulative staking rewards over time for different NFT levels. Higher levels yield proportionally greater returns, with APB providing additional vintage bonus.

6 Migration System

6.1 Token Migration

The protocol supports migration from previous contract versions through the Migratable base contract. Users can convert old tokens to new tokens within a deadline:

$$\text{deadline} = t_{\text{deploy}} + \Delta t_{\text{migration}} \tag{23}$$

The current migration window is approximately 4 years (126,230,400 seconds).

6.2 NFT Migration

NFT migration uses a two-step process:

1. Burn old NFTs from previous contract
2. Deposit equivalent XPOW (potentially migrated) to mint new NFTs

The system attempts multiple level variations to find matching old NFTs, handling cases where level encoding changed between versions.

6.3 Atomic SOV Migration

APower migration is atomic: burning old APOW triggers migration of the corresponding XPOW backing, then mints new APOW with the migrated XPOW as collateral.

7 Security Considerations

7.1 Security Audit

The XPower smart contracts underwent a comprehensive security audit by Hacken.io in January 2024 [11]. The audit achieved an overall score of 9.1/10, with:

- Security Score: 10/10
- Code Quality Score: 9/10
- Test Coverage: 80.31%
- Documentation Quality Score: 10/10

Six findings were identified and all were resolved:

- **Critical (1)**: Block hash validity window—the `_recent` function was corrected to properly compare timestamps, ensuring block hashes expire after the intended one-hour interval
- **High (2)**: NFT transfer age tracking and reward accumulation issues were resolved by implementing proportional age adjustment in `_pushBurn` and adding `refreshClaimed` functionality
- **Low (2)**: Migration access controls were strengthened to ensure voluntary migration

7.2 Anti-Gaming: Banq System

The Banq library implements several anti-gaming protections:

1. **Minimum balance ratio**: Claims must satisfy

$$\frac{\text{netBalance}}{\text{excessBalance}} > \text{MIN\_NET2RIP} \tag{24}$$

   where MIN_NET2RIP = 100. Excess balance is burned, preventing claim inflation.

2. **Supply cap**: Without a lending pool, total APOW supply is capped at 43,830 tokens.
3. **Pool locking**: With a pool configured, claimed tokens are supplied and locked, requiring additional PoW for withdrawal.

7.3 Reentrancy Protection

Critical functions use OpenZeppelin's ReentrancyGuardTransient, which leverages EIP-1153 transient storage for gas-efficient reentrancy prevention.

7.4 Access Control

Role-based access control restricts sensitive operations:

- `MOE_SEAL_ROLE`: Seal XPOW migration
- `SOV_SEAL_ROLE`: Seal APOW migration
- `APR_ROLE`: Modify APR parameters
- `APB_ROLE`: Modify APB parameters

7.5 Reparametrization Limits

The Rpp library constrains parameter changes:

- Value changes limited to $\pm 10\%$ per update
- Minimum 1 month between updates
- Array structure validation

7.6 Block Hash Freshness

The 1-hour block hash validity window limits precomputation attacks while providing sufficient time for nonce submission during network congestion.

7.7 Known Risks

The security audit identified several inherent risks that users should acknowledge [11]:

1. **APOW Supply Variation**: Actual minting amounts may deviate from projections based on claiming frequency. Frequent early claims can accelerate token supply growth.
2. **Migration Deadlines**: Token migration is subject to deadlines and can be sealed by authorized roles, potentially impacting late migrators.
3. **Dynamic Rate Impact**: Staking higher-level NFTs significantly affects rewards for all levels through the auto-adjustment mechanism.
4. **APR/APB Parameter Control**: Administrative roles can modify reward parameters within bounded constraints, though changes become increasingly difficult over time.
5. **Transfer Reward Reset**: Transferring APowerNfts (staked positions) resets accumulated but unclaimed rewards. Users should claim before transfers.

8 Conclusion

XPower presents a novel approach to proof-of-work token distribution that separates mining from consensus, enabling accessible participation without specialized hardware. The NFT staking system provides flexible position management, while the time-weighted reward mechanism incentivizes long-term commitment. The polynomial-based rate system and dynamic adjustment create a self-balancing ecosystem that rewards diverse staking strategies.

The protocol has undergone professional security auditing by Hacken.io, achieving a score of 9.1/10 with all identified vulnerabilities resolved [11]. Key security measures include reentrancy protection via OpenZeppelin's ReentrancyGuardTransient, role-based access control, and the Banq anti-gaming system.

Future work includes analysis of mining difficulty distribution, optimization of reward parameters based on observed staking patterns, and integration with decentralized lending protocols for the Banq system.

Acknowledgments

The XPower protocol builds on the OpenZeppelin contracts library (v5.3.0) and PRB Math (v4.1.0) for fixed-point arithmetic. The implementation is deployed on Avalanche C-Chain. Security auditing was performed by Hacken.io, whose thorough review identified and helped resolve critical vulnerabilities in the block hash validation and staking reward systems.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[2] V. Buterin, "Ethereum: A next-generation smart contract and decentralized application platform," 2014. [Online]. Available: https://ethereum.org/whitepaper/

[3] tevador et al., "RandomX: ASIC-resistant proof-of-work algorithm," 2019. [Online]. Available: https://github.com/tevador/RandomX

[4] R. Leshner and G. Hayes, "Compound: The money market protocol," 2019. [Online]. Available: https://compound.finance/documents/Compound.Whitepaper.pdf

[5] Lido, "Lido: Decentralized staking for Ethereum," 2020. [Online]. Available: https://lido.fi/static/Lido:Ethereum-Liquid-Staking.pdf

[6] M. Egorov, "StableSwap - efficient mechanism for Stablecoin liquidity," 2019. [Online]. Available: https://resources.curve.finance/pdf/curve-stableswap.pdf

[7] W. Radomski, A. Cooke, P. Castonguay, J. Therien, E. Binet, and R. Sandford, "EIP-1155: Multi Token Standard," Ethereum Improvement Proposals, 2018. [Online]. Available: https://eips.ethereum.org/EIPS/eip-1155

[8] K. Sekniqi, D. Laine, S. Buttolph, and E. G. Sirer, "Avalanche Platform," Ava Labs, 2020. [Online]. Available: https://www.avax.network/whitepapers

[9] OpenZeppelin, "OpenZeppelin Contracts," 2023. [Online]. Available: https://docs.openzeppelin.com/contracts

[10] P. R. Barber, "PRB Math: Solidity library for fixed-point math," 2023. [Online]. Available: https://github.com/PaulRBerg/prb-math

[11] Hacken.io, "Smart Contract Code Review and Security Analysis Report for XPower," January 2024. Commit: 272f69a. Overall Score: 9.1/10.

A Glossary

APB Annual Percentage Bonus. Additional reward rate based on NFT vintage year.

APower (APOW) Aged Power token. A rate-limited store-of-value token backed by XPOW.

APowerNft ERC-1155 tokens representing staked XPowerNft positions with age tracking.

APR Annual Percentage Rate. The base reward rate per NFT level.

Banq The auto-supply and anti-gaming library for APOW claims.

Block Hash A 32-byte hash used as mining reference, valid for one hour.

Denomination The XPOW value represented by an NFT level (10^{ℓ} tokens).

Integrator A mechanism for computing duration-weighted arithmetic means of rate values.

Level NFT classification (0, 3, 6, ..., 99) determining denomination.

Migration The process of converting tokens from old contract versions.

MoeTreasury The contract managing staking rewards and APR/APB parameters.

Nibble A half-byte (4 bits); one hexadecimal digit.

Nonce A value discovered through mining that produces a valid hash.

NftTreasury The contract managing XPowerNft staking and APowerNft minting.

Polynomial Mathematical function used for APR/APB calculation with format $[add, div, mul, exp]$.

Proof-of-Work (PoW) A mechanism requiring computational effort to produce valid tokens.

Shares Accumulator tracking total staked value per NFT level for rate adjustment.

XPower (XPOW) The base proof-of-work ERC-20 token.

XPowerNft ERC-1155 tokens representing deposited XPOW with level and year encoding.

Year The calendar year component of an NFT identifier, starting from 2021.

Zeros The number of leading zero nibbles in a hash, determining mining difficulty.

Table 3: Mainnet deployment (Avalanche C-Chain, ID: 43114)

Contract	Abbreviation
XPower	MOE
XPowerNft	NFT
APowerNft	PPT
APower	SOV
MoeTreasury	MTY
NftTreasury	NTY

B Contract Addresses

Contract addresses are stored in environment files and can be verified on Snowtrace.

C Mathematical Derivations

C.1 Expected Token Yield

For a miner performing N hash attempts, the expected token yield is:

$$E[\text{tokens}] = N \sum_{z=1}^{64} \frac{2^z - 1}{16^z} = N \sum_{z=1}^{64} \frac{2^z - 1}{2^{4z}} \tag{25}$$

Simplifying:

$$E[\text{tokens}] = N \left(\sum_{z=1}^{64} 2^{-3z} - \sum_{z=1}^{64} 2^{-4z} \right) \tag{26}$$

Using geometric series:

$$E[\text{tokens}] \approx N \left(\frac{1/8}{1 - 1/8} - \frac{1/16}{1 - 1/16} \right) \tag{27}$$

$$= N \left(\frac{1}{7} - \frac{1}{15} \right) \approx 0.076N \tag{28}$$

Thus, approximately one token is expected per 13 hash attempts on average.

C.2 Staking Age Invariant

The age tracking maintains the invariant that partial unstaking proportionally reduces age:

$$\text{age}_{\text{effective}} = \text{balance} \times t - A \tag{29}$$

After burning amount b from balance B:

$$A' = A - A \times \frac{b}{B} = A \times \frac{B-b}{B} \tag{30}$$

$$\begin{aligned} \text{age}'_{\text{effective}} &= (B-b) \times t - A' && (31)\\ &= (B-b) \times t - A \times \frac{B-b}{B} && (32)\\ &= (B-b)\left(t - \frac{A}{B}\right) && (33)\\ &= \frac{B-b}{B} \times \text{age}_{\text{effective}} && (34) \end{aligned}$$

This preserves the average staking duration per remaining token.