Code Overview

1. Minting: mintToken.js (589M $DRIPPY, done).
2. Distribution: distributeTokens.js (58.9M LP, etc.).
3. Escrow: setupEscrow.js (484.747M locked).
4. Hook: drippy\_hook.c (1% NFT rewards in XRP, 50% sell tax).
5. Bridge: bridgeTokens.js (XRPL Xahau).
6. Liquidity: addLiquidity.js (58.9M + $10K XRP).
7. Anti-Sniping: antiSnipingDecoys.js (5 decoys).
8. NFT Rewards: batchNFTRewards.js (XRP batching).
9. Relocking: relockUnused.js.

Compile: wasi-sdk for .c, Node.js for .js. Deploy: SetHook via Xaman.

$DRIPPY Rundown

* Token: 589M $DRIPPY, minted August 13.
* Bridge: Complete.
* Hooks: 5% fee (1% NFTs in XRP, 2% holders, etc.).
* Escrow: 484.747M locked.
* NFTs: Minted
* Distribution: 58.9M LP, 15.9M founders, 29.45M Treasury #2, 484.747M escrow.
* Wallets: Issuer (TBD), Deployer (rwprJf1...), AMM (rae3QB8...), Team (rJVUU12q..., confirm), Treasury #2 (rGuAGX7...), Escrow (rGNZ4e...).

1. Token MintingMint 589M $DRIPPY from Issuer Wallet (2-of-3 multisig).

javascript

// File: mintToken.js

const { Client, Wallet } = require('xrpl');

async function mintToken() {

const client = new Client('wss://s1.ripple.com'); // XRPL mainnet

await client.connect();

const teamWallet = Wallet.fromSeed('team\_signer\_seed'); // Team signer 1 (replace)

const tx = {

TransactionType: 'Payment',

Account: 'issuer\_r\_address', // Issuer Wallet (replace)

Destination: 'operational\_r\_address', // Operational Wallet (replace)

Amount: {

currency: 'DRIPPY',

value: '589000000', // 589M $DRIPPY

issuer: 'issuer\_r\_address',

},

};

const result = await client.submit(tx, { wallet: teamWallet, autofill: true }); // Multisig via Xaman

console.log('Minting Submitted:', result);

await client.disconnect();

}

// Run: node mintToken.js

// Cost: ~0.000012 XRP (~$0)

// Time: 3-5 seconds

// Status: Complete (August 13, 2025)

Distribute: 58.9M to Operational (LP), 15.9M to Team (locked), 29.45M to Treasury #2, 484.747M to Escrow.

javascript

// File: distributeTokens.js

const { Client, Wallet } = require('xrpl');

async function distributeTokens() {

const client = new Client('wss://s1.ripple.com');

await client.connect();

const issuerWallet = Wallet.fromSeed('issuer\_seed'); // Issuer signer (replace)

const distributions = [

{ destination: 'operational\_r\_address', amount: '58900000' }, // LP

{ destination: 'rJVUU12qUKFGccukM9Cmr6dZewyUshZY8K', amount: '15903000' }, // Team lock

{ destination: 'rGuAGX7XTJA8N84ChScxrpQW9CT37VQkoZ', amount: '29450000' }, // Treasury #2

{ destination: 'rGNZ4eXcwqCjcqAR9RWX5vcDJ8wH1upLZB', amount: '484747000' }, // Escrow

];

for (const dist of distributions) {

const tx = {

TransactionType: 'Payment',

Account: 'issuer\_r\_address',

Destination: dist.destination,

Amount: {

currency: 'DRIPPY',

value: dist.amount,

issuer: 'issuer\_r\_address',

},

};

const result = await client.submit(tx, { wallet: issuerWallet, autofill: true });

console.log('Distributed to', dist.destination, ':', result);

}

await client.disconnect();

}

// Run: node distributeTokens.js

// Cost: ~0.000048 XRP (4 txns, ~$0)

// Time: 12-20 seconds

// Status: Complete

Escrow Setup and Time-Locked ReleasesLock 484.747M $DRIPPY, set 2% monthly releases (1,767,000 $DRIPPY/month, 5% Team, 95% Deployer).

javascript

// File: setupEscrow.js

const { Client, Wallet } = require('xrpl');

async function setupEscrow() {

const client = new Client('wss://s1.ripple.com');

await client.connect();

const issuerWallet = Wallet.fromSeed('issuer\_seed');

const nonCirculating = '484747000.000000';

const escrowTx = {

TransactionType: 'EscrowCreate',

Account: 'issuer\_r\_address',

Destination: 'rGNZ4eXcwqCjcqAR9RWX5vcDJ8wH1upLZB',

Amount: nonCirculating,

FinishAfter: Math.floor(Date.now() / 1000) + 31536000, // 1-year lock

};

await client.submit(escrowTx, { wallet: issuerWallet, autofill: true });

// Monthly releases (12 months example)

for (let month = 1; month <= 12; month++) {

const monthly = '1767000.000000'; // 2% of 88.35M

const monthlyTx = {

TransactionType: 'EscrowCreate',

Account: 'issuer\_r\_address',

Destination: 'rGNZ4eXcwqCjcqAR9RWX5vcDJ8wH1upLZB',

Amount: monthly,

FinishAfter: Math.floor(Date.now() / 1000) + (month \* 2592000), // 30 days

};

await client.submit(monthlyTx, { wallet: issuerWallet, autofill: true });

}

await client.disconnect();

}

// Run: node setupEscrow.js

// Cost: ~0.1 XAH (~$1.3 for 13 txns)

// Time: 39-65 seconds

// Status: Complete

Hook Logic with NFT Rewards in XRP and Anti-SnipingThe drippy\_hook.wasm enforces the 5% fee (1% NFTs in XRP, 2% holders, 1% Treasury, 1% AMM, 0.5% XRP LP, 0.5% $DRIPPY LP) and a 50% sell tax for 1 hour (e.g., 6-7 PM WAT, September 11, 2025). NFT rewards are converted to XRP using AMM price.

c

// File: drippy\_hook.c

#include <xrpl.h>

int64\_t hook(int64\_t reserved) {

// Check transaction type

int64\_t tx\_type = tx\_type();

if (tx\_type != BUY && tx\_type != SELL) return 0;

// Anti-sniping: 50% sell tax for 1 hour

int64\_t amount = tx\_amount();

if (ledger\_time() < 1726150800 + 3600 && tx\_type == SELL) { // 6-7 PM WAT, Sep 11

if (!is\_whitelisted(tx\_account(), "issuer\_r\_address,rae3QB8qYkseWuNmkmHGXQJHFybSQ3115J,rwprJf1ZEU3foKSiwhDg5kj9zDWFtPgMqJ")) {

int64\_t fee = amount \* 0.5; // 50% sell tax

transfer("rwprJf1ZEU3foKSiwhDg5kj9zDWFtPgMqJ", fee, "DRIPPY", "issuer\_r\_address");

amount -= fee;

}

}

// Normal 5% fee

int64\_t fee = amount \* 0.05;

int64\_t nft\_reward\_drippy = fee \* 0.01; // 1% for NFT holders

int64\_t holder\_reward = fee \* 0.02; // 2% token holders

int64\_t treasury = fee \* 0.01; // 1% Treasury

int64\_t amm\_deposit = fee \* 0.01; // 1% AMM

int64\_t xrp\_lp = fee \* 0.005; // 0.5% XRP LP

int64\_t drippy\_lp = fee \* 0.005; // 0.5% $DRIPPY LP

// Convert NFT reward to XRP (using AMM price, ~$0.00016966)

int64\_t nft\_reward\_xrp = (nft\_reward\_drippy \* 16966) / 1000000; // Convert $DRIPPY to XRP drops

int64\_t total\_nfts = 0;

int64\_t holders[100]; // Max 100 holders

int64\_t nft\_counts[100];

int64\_t holder\_count = query\_nft\_holders(holders, nft\_counts, 5); // NFTokenPage

for (int i = 0; i < holder\_count; i++) {

total\_nfts += nft\_counts[i];

}

for (int i = 0; i < holder\_count; i++) {

int64\_t share = (nft\_reward\_xrp \* nft\_counts[i]) / total\_nfts;

if (share > 0) {

transfer(holders[i], share, "XRP", NULL); // Pay in XRP drops

}

}

// Other distributions

distribute\_holders(holder\_reward); // Proportional in $DRIPPY

transfer("rwprJf1ZEU3foKSiwhDg5kj9zDWFtPgMqJ", treasury, "DRIPPY", "issuer\_r\_address");

amm\_deposit\_function("rae3QB8qYkseWuNmkmHGXQJHFybSQ3115J", amm\_deposit);

lp\_inject("xrp\_lp\_wallet", xrp\_lp, "DRIPPY", "issuer\_r\_address");

lp\_inject("drippy\_lp\_wallet", drippy\_lp, "DRIPPY", "issuer\_r\_address");

return 0;

}

// Compile: wasi-sdk clang --target=wasm32-wasi drippy\_hook.c -o drippy\_hook.wasm

// Deploy: SetHook via 2-of-3 multisig (Xaman)

Cross-Chain Bridge (XRPL Xahau)Bridge $DRIPPY with 0.1 XAH burn on Xahau, fallback fee on XRPL.

javascript

// File: bridgeTokens.js

const { Client, Wallet } = require('xrpl');

async function bridgeTokens(fromNetwork, toNetwork, amount, senderAddress) {

const clientFrom = new Client(fromNetwork === 'XRPL' ? 'wss://s1.ripple.com' : 'wss://xahau.network');

const clientTo = new Client(toNetwork === 'XRPL' ? 'wss://s1.ripple.com' : 'wss://xahau.network');

await clientFrom.connect();

await clientTo.connect();

const wallet = Wallet.fromSeed('sender\_seed');

// Burn 0.1 XAH on Xahau

if (fromNetwork === 'Xahau') {

const burnTx = {

TransactionType: 'Payment',

Account: senderAddress,

Destination: 'rHb9CJAWyB4rj91VRWn96DkukG4bwdtyTh',

Amount: '100000', // 0.1 XAH

};

await clientFrom.submit(burnTx, { wallet, autofill: true });

}

// Transfer $DRIPPY

const bridgeTx = {

TransactionType: 'Payment',

Account: senderAddress,

Destination: 'bridge\_r\_address', // Bridge Wallet (replace)

Amount: {

currency: 'DRIPPY',

value: amount.toString(),

issuer: 'issuer\_r\_address',

},

};

const result = await clientFrom.submit(bridgeTx, { wallet, autofill: true });

console.log('Bridge Transfer:', result);

await clientTo.disconnect();

await clientFrom.disconnect();

}

// Run: node bridgeTokens.js 'XRPL' 'Xahau' 1000 'sender\_r\_address'

// Cost: ~0.1 XAH + 0.000012 XRP (~$0.01)

// Time: 10-15 seconds

// Status: Complete

Liquidity AdditionAdd 58.9M $DRIPPY + $10K XRP, burn 5k LP tokens.

javascript

// File: addLiquidity.js

const { Client, Wallet } = require('xrpl');

async function addLiquidity() {

const client = new Client('wss://s1.ripple.com');

await client.connect();

const wallet = Wallet.fromSeed('deployer\_seed');

const tx = {

TransactionType: 'AMMDeposit',

Account: 'rwprJf1ZEU3foKSiwhDg5kj9zDWFtPgMqJ',

Asset: { currency: 'DRIPPY', issuer: 'issuer\_r\_address' },

Amount: '58900000.000000',

Asset2: { currency: 'XRP' },

Amount2: '10000000000', // $10K XRP

};

const result = await client.submit(tx, { wallet, autofill: true });

console.log('Liquidity Added:', result);

// Burn 5k LP tokens

const burnTx = {

TransactionType: 'Payment',

Account: 'rwprJf1ZEU3foKSiwhDg5kj9zDWFtPgMqJ',

Destination: 'rHb9CJAWyB4rj91VRWn96DkukG4bwdtyTh',

Amount: { currency: 'LPTOKEN', value: '5000', issuer: 'rae3QB8qYkseWuNmkmHGXQJHFybSQ3115J' },

};

await client.submit(burnTx, { wallet, autofill: true });

await client.disconnect();

}

// Run: node addLiquidity.js

// Cost: ~$10

// Time: 3-5 seconds

// Status: Pending (confirm)

Off-Chain NFT Reward Batching in XRPBatch daily NFT rewards in XRP to save ~90% on fees.

javascript

// File: batchNFTRewards.js

const { Client, Wallet } = require('xrpl');

async function batchNFTRewards() {

const client = new Client('wss://s1.ripple.com');

await client.connect();

const treasuryWallet = Wallet.fromSeed('treasury\_seed');

// Query NFT holders (5+ NFTs)

const nftHolders = await fetchNFTHolders('issuer\_r\_address', 5);

const totalNFTs = nftHolders.reduce((sum, h) => sum + h.nftCount, 0);

const dailyRewards = await fetchDailyRewards(); // Assume 100 $DRIPPY daily

// Convert to XRP (~$0.00016966 per $DRIPPY)

const nftRewardXRP = (dailyRewards \* 0.01 \* 16966) / 1000000; // 1% in XRP drops

for (const holder of nftHolders) {

const share = (nftRewardXRP \* holder.nftCount) / totalNFTs;

if (share > 0) {

const tx = {

TransactionType: 'Payment',

Account: 'rwprJf1ZEU3foKSiwhDg5kj9zDWFtPgMqJ',

Destination: holder.address,

Amount: share.toString(), // XRP drops

};

await client.submit(tx, { wallet: treasuryWallet, autofill: true });

console.log(`Sent ${share / 1000000} XRP to ${holder.address}`);

}

}

await client.disconnect();

}

async function fetchNFTHolders(issuer, minNFTs) {

// Use account\_nfts RPC or bithomp API

return [

{ address: 'rHolder1...', nftCount: 10 },

{ address: 'rHolder2...', nftCount: 5 },

].filter(h => h.nftCount >= minNFTs);

}

async function fetchDailyRewards() {

// Query hook state or ledger

return 100; // Placeholder

}

// Run: node batchNFTRewards.js (daily via cron/AWS Lambda)

// Cost: ~0.000012 XRP per batch (~$0)

Relock unused $DRIPPY.

javascript

// File: relockUnused.js

const { Client, Wallet } = require('xrpl');

async function relockUnused() {

const client = new Client('wss://s1.ripple.com');

await client.connect();

const wallet = Wallet.fromSeed('team\_signer\_seed');

const tx = {

TransactionType: 'EscrowCreate',

Account: 'rJVUU12qUKFGccukM9Cmr6dZewyUshZY8K',

Destination: 'rGNZ4eXcwqCjcqAR9RWX5vcDJ8wH1upLZB',

Amount: '50000.000000',

FinishAfter: Math.floor(Date.now() / 1000) + 31536000, // 1-year lock

};

const result = await client.submit(tx, { wallet, autofill: true });

console.log('Relocked:', result);

await client.disconnect();

}

// Run: node relockUnused.js

// Cost: ~0.1 XAH

// Time: 3-5 seconds