

# DAPA Wallet Setup and Requirements Guide

## Overview

DAPA is a privacy-focused cryptocurrency built with BlockDAG, Homomorphic Encryption, Zero-Knowledge Proofs, and Smart Contracts. This guide covers building and running a DAPA wallet from source.

## System Requirements

### Hardware Requirements

- **RAM:** Minimum 4GB, Recommended 8GB+
- **Storage:** 50GB+ free space (for blockchain data and precomputed tables)
- **CPU:** Multi-core processor (for compilation and wallet operations)

- **Network:** Stable internet connection for blockchain synchronization

## Software Requirements

- **Operating System:** Linux (Ubuntu 20.04+ recommended)
- **Rust:** Latest stable version (1.70+)
- **Build Tools:** git, cmake, clang, pkg-config
- **Dependencies:** OpenSSL development libraries

## Installation Prerequisites

### 1. Install Rust

```
bash
```

```
curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
```

```
source ~/.cargo/env
```

```
rustup update stable
```

## 2. Install Build Dependencies (Ubuntu/Debian)

```
bash
```

```
sudo apt update
```

```
sudo apt install -y \
```

```
git \
```

```
cmake \
```

```
clang \
```

```
pkg-config \
```

```
libssl-dev \
```

```
build-essential \
```

```
curl
```

## 3. Install Build Dependencies (CentOS/RHEL)

```
bash
```

```
sudo yum groupinstall "Development Tools"
```

```
sudo yum install -y git cmake clang openssl-devel
```

# Building DAPA from Source

## 1. Clone the Repository

```
bash
```

```
git clone https://github.com/dapahe/dapa-blockchain.git  
cd dapa-blockchain
```

## 2. Build the Project

```
bash
```

```
# Build all components (daemon, wallet, miner)
```

```
cargo build --release
```

```
# Or build specific components
```

```
cargo build --release --bin dapa_wallet
```

```
cargo build --release --bin dapa_daemon
```

```
cargo build --release --bin dapa_miner
```

### 3. Verify Build

After successful compilation, binaries will be in:

```
bash
```

```
ls -la target/release/
```

```
# Should show: dapa_daemon, dapa_wallet, dapa_miner
```

## Initial Setup

### 1. Create Directory Structure

```
bash
```

```
mkdir -p ~/dapa-node/{wallets,logs,data}
```

```
cd ~/dapa-node
```

### 2. Copy Binaries

```
bash
```

```
cp /path/to/dapa-blockchain/target/release/dapa_* ~/d
```

### 3. Generate Precomputed Tables

```
bash
```

*# This can take significant time and disk space*

```
./dapa_wallet --precomputed-tables-l1 26
```

## Wallet Configuration

### 1. Create Wallet Credentials File

Create `wallet_credentials.json`:

```
json
```

```
{  
  "wallet_1": {  
    "username": "your_username",  
    "password": "your_secure_password"  
  },  
  "wallet_2": {  
    "username": "another_username",  
    "password": "another_secure_password"  
  }  
}
```

## 2. Wallet Creation Methods

### Interactive Creation

```
bash  
  
./dapa_wallet  
  
# At prompt:  
# > create  
  
# Follow prompts for wallet name and password
```

# Command Line Parameters

```
bash
```

```
./dapa_wallet --wallet-path ./wallets/wallet_1 --password
```

## Network Configuration

### 1. Mainnet Configuration

```
bash
```

```
./dapa_wallet --network mainnet --daemon-address http
```

### 2. Testnet Configuration

```
bash
```

```
./dapa_wallet --network testnet --daemon-address http;
```



## 3. Local Development

```
bash
```

```
# Start local daemon first
```

```
./dapa_daemon --network dev
```

```
# Then connect wallet
```

```
./dapa_wallet --network dev --daemon-address http://127.0.0.1:8081
```



## RPC Server Setup

### 1. Start RPC Server from CLI

```
bash
```

```
# In wallet CLI
```

```
> start_rpc_server 127.0.0.1:8081 username password
```

### 2. Start with RPC Enabled

```
bash
```

```
./dapa_wallet \  
--rpc-bind-address "127.0.0.1:8081" \  
--rpc-username "admin" \  
--rpc-password "secure_password" \  
--wallet-path ./wallets/my_wallet \  
--password "wallet_password"
```

# Essential Wallet Operations

## 1. Basic Commands (Interactive Mode)

```
bash
```

## *# Wallet management*

**open** *# Open existing wallet*

**create** *# Create new wallet*

**logout** *# Close current wallet*

## *# Transaction operations*

**transfer** *# Send funds*

**transfer\_all** *# Send all balance*

**balance** *# Check balance*

**history** *# View transaction history*

## *# Wallet information*

**display\_address** *# Show wallet address*

**nonce** *# Show current nonce*

**seed** *# Show recovery seed*

## *# Network operations*

**online\_mode** *# Connect to daemon*

**offline\_mode** *# Disconnect from daemon*

**rescan** *# Resync wallet data*

## **2. Transaction Example**

bash

*# In wallet CLI:*

> transfer

Enter destination address: dap:address\_here

Enter amount: 1.5

Enter asset (default DAPA): [press enter]

# Important Files and Directories

## Directory Structure

```
dapa-node/
├── dapa_wallet          # Wallet executable
├── dapa_daemon          # Daemon executable
├── dapa_miner           # Miner executable
├── wallets/            # Wallet storage
│   ├── wallet_1/
│   └── wallet_2/
├── logs/                # Application logs
└── precomputed_tables_26.bin # Precomputed tables
```

└─ wallet\_credentials.json # Automation  
credentials

## Critical Files

- **Wallet Files:** `wallets/*/` - Contains encrypted wallet data
- **Precomputed Tables:** `precomputed_tables_*.bin`  
- Required for operations
- **Logs:** `logs/` - Debugging and operational logs
- **Config:** Various JSON configuration files

## Security Considerations

### 1. Wallet Security

- Use strong, unique passwords for each wallet
- Backup wallet seed phrases securely
- Store wallet files in encrypted directories
- Never share private keys or seeds

## 2. RPC Security

- Bind RPC only to localhost unless necessary
- Use strong RPC authentication credentials
- Consider firewall rules for RPC ports
- Monitor RPC access logs

## 3. Network Security

- Verify daemon addresses before connecting
- Use HTTPS endpoints when available
- Monitor network connections
- Keep software updated

# Troubleshooting

## Common Issues

### 1. Build Failures

```
bash
```

*# Update Rust*

rustup update stable

*# Clear cache and rebuild*

cargo clean

cargo build --release

## 2. Missing Dependencies

bash

*# Ubuntu/Debian*

`sudo apt install libssl-dev pkg-config cmake clang`

*# Check Rust installation*

rustc --version

cargo --version

## 3. Wallet Connection Issues

bash

*# Check daemon connectivity*

`curl http://node.dapahe.com:20101`

*# Verify wallet files exist*

`ls -la wallets/`

*# Check logs*

`tail -f logs/dapa-wallet.log`

## 4. RPC Authentication Errors

- Verify username/password in credentials file
- Check RPC bind address and port
- Ensure wallet is properly opened before RPC calls

## Log Analysis

bash



*# Real-time log monitoring*

`tail -f logs/dapa-wallet.log`

*# Search for errors*

`grep -i error logs/dapa-wallet.log`

*# Filter by date*

`grep "2024-09-18" logs/dapa-wallet.log`

## Automation Scripts

### 1. Prerequisites for Automation

- Python 3.7+
- Working DAPA wallet CLI
- Proper wallet credentials configuration

### 2. Example Automation Structure

python

*# Control CLI programmatically*

```
import subprocess
```

```
import time
```

*# Start wallet process*

```
process = subprocess.Popen(['./dapa_wallet'],  
                             stdin=subprocess.PIPE,  
                             stdout=subprocess.PIPE,  
                             text=True)
```

*# Send commands*

```
process.stdin.write('open\n')
```

```
process.stdin.write('wallet_name\n')
```

```
process.stdin.write('password\n')
```

# Performance Optimization

## 1. Precomputed Tables

- L1=26: Full performance, ~350MB file
- L1=18: Medium performance, ~2MB file
- L1=13: Low performance, ~64KB file

## 2. System Optimization

```
bash
```

```
# Increase file descriptor limits
```

```
ulimit -n 65536
```

```
# Optimize for SSD storage
```

```
echo noop | sudo tee /sys/block/sda/queue/scheduler
```

## Maintenance

### 1. Regular Tasks

- Monitor disk space for blockchain data
- Backup wallet files regularly
- Update software when new versions release
- Review and rotate logs

## 2. Backup Strategy

```
bash
```

```
# Backup wallet files
```

```
tar -czf wallet_backup_$(date +%Y%m%d).tar.gz wallets/
```

```
# Backup configuration
```

```
cp wallet_credentials.json config_backup.json
```

```
# Store seed phrases securely offline
```

## Support and Resources

### Official Resources

- Documentation: <https://docs.dapahe.com/>
- GitHub: <https://github.com/dapahe/dapa-blockchain>
- Community: DAPA official channels

## Development Environment

- Use testnet for development and testing
  - Never test with mainnet funds
  - Monitor resource usage during development
  - Keep development and production environments separate
- 

## Notes

This guide is based on analysis of DAPA blockchain source code and operational testing. Requirements may vary based on specific use cases and network conditions. Always verify compatibility with the latest DAPA releases.

The DAPA blockchain uses advanced cryptographic features that require significant computational resources. Plan system resources accordingly for optimal performance.