Starknet Lightning Privacy Mixer - Technical Specification

System Overview

The Privacy Mixer breaks on-chain linkability between Starknet accounts by routing funds through Bitcoin Lightning Network and Cashu e-cash systems.

Flow: STRK (Account A) → Lightning BTC → Cashu e-cash → Lightning BTC → STRK (Account B)

Architecture Components

1. Starknet Layer

- **Smart Contract**: Cairo contract managing mixing operations and state
- **Wallet Integration**: Support for ArgentX, Braavos wallets
- **Account Management**: Handle multiple destination accounts

2. Lightning Network Integration

- **Payment Processing**: Invoice generation and payment handling
- **Swap Interface**: Integration with Atomiq for STRK ↔ Lightning BTC
- **Error Handling**: Timeout and failure recovery mechanisms

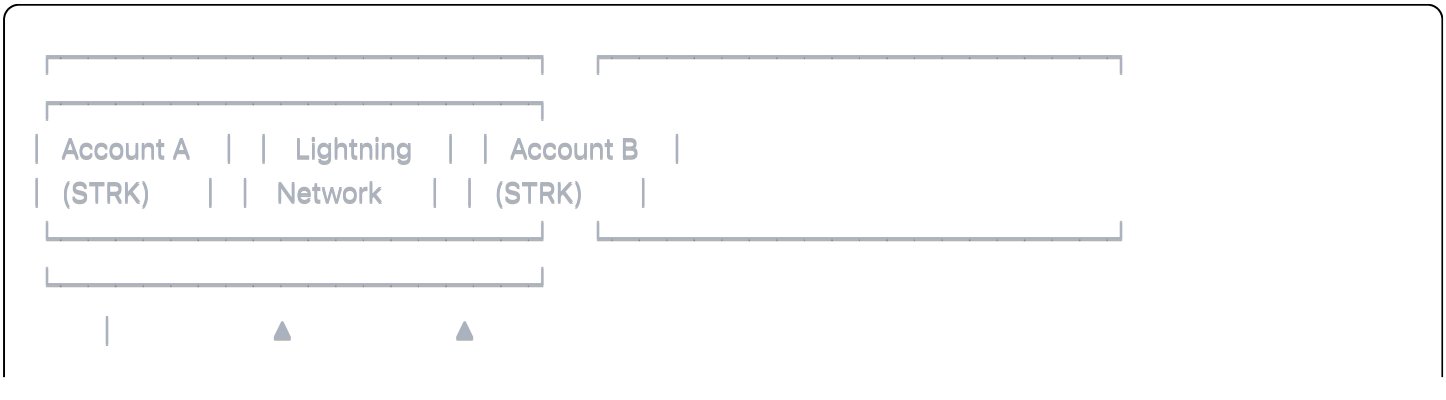
3. Cashu E-cash Privacy Layer

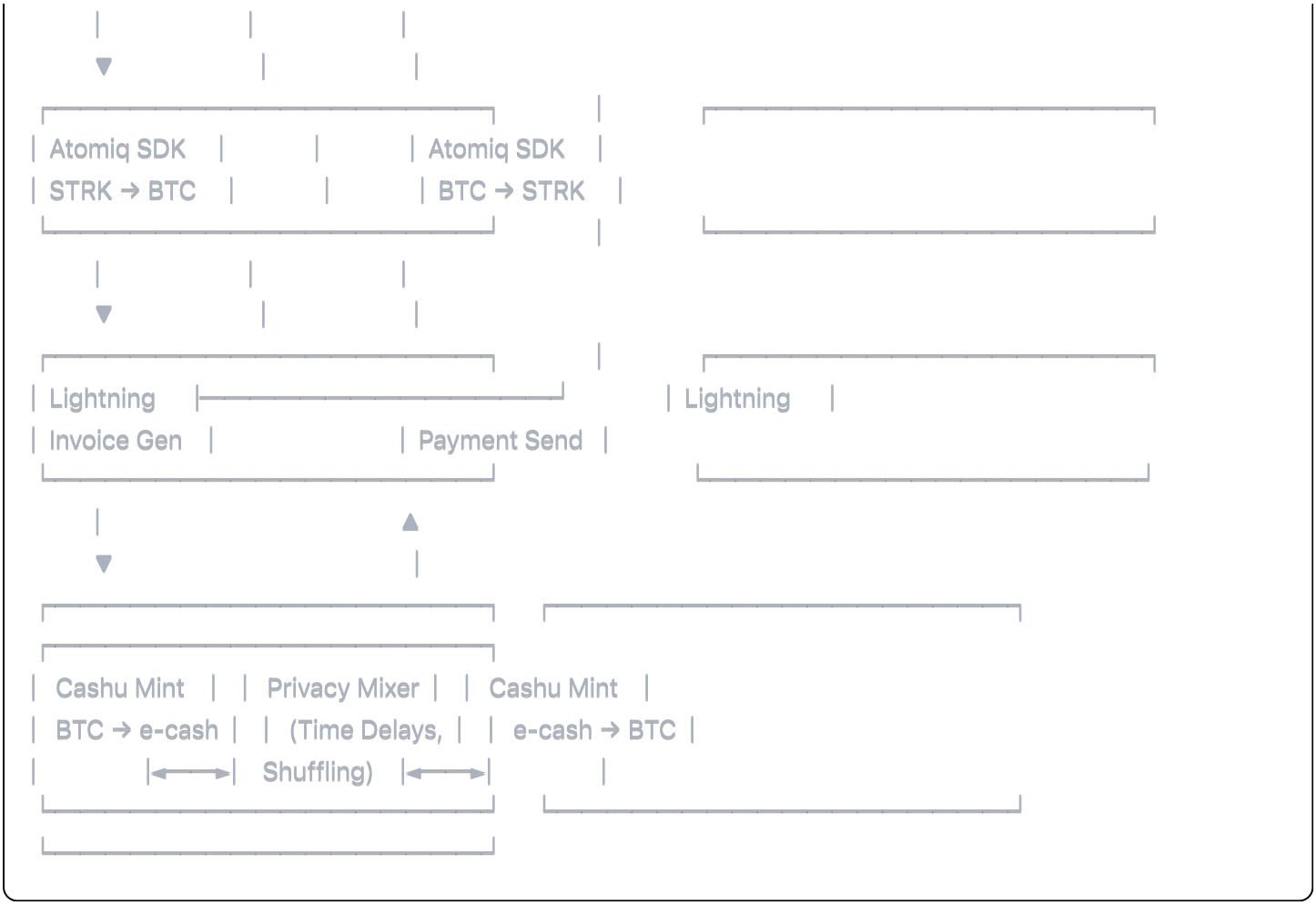
- **Token Minting**: Convert Lightning BTC to Cashu e-cash "nuts"
- **Privacy Mixing**: Split, shuffle, and recombine tokens
- **Token Redemption**: Convert e-cash back to Lightning BTC

4. Privacy Enhancements

- **Time Delays**: Configurable delays between operations
- **Amount Obfuscation**: Split outputs across multiple wallets
- **Routing Diversification**: Use multiple Cashu mints

Technical Architecture Diagram





Key Features

Privacy Guarantees

- **Unlinkability**: No direct on-chain connection between input/output accounts
- **Anonymous Set**: Mixed with other users' transactions
- **Temporal Privacy**: Time delays prevent timing analysis

Performance Targets

- **Speed**: < 10 minutes total mixing time
- **Fees**: < 2% total transaction cost
- **Reliability**: > 99% success rate

Security Measures

- **Non-custodial**: No intermediary holds user funds
- **Atomic Operations**: All-or-nothing transaction semantics
- **Timeout Protection**: Automatic refund mechanisms

Integration Patterns

Atomiq SDK Integration

- Use existing Atomiq integration for STRK ↔ Lightning BTC swaps that went live on Braavos in April 2025, enabling users to scan QR codes at Lightning-accepting merchants and pay with STRK while merchants receive BTC instantly
- Leverage zero slippage atomic swaps with minimal counterparty risk through on-chain escrow mechanism using Bitcoin's PoW consensus
- Handle swap state management and error recovery

Cashu E-cash Integration

- Integrate Cashu protocol for instant, nearly free e-cash transactions with bearer instrument "nuts" backed by full bitcoin reserve
- Utilize mint selection for privacy enhancement - users can choose between multiple mints
- Implement Lightning interoperability for instant withdraw and privacy mixing through token splitting

Lightning Network Patterns

- Invoice-based payment flow
- Multi-path payment routing
- Channel liquidity management
- Payment failure handling

Data Flow

1. **Input Phase**
   - User initiates mix from Account A with STRK amount
   - Smart contract validates and locks funds
   - Generate Lightning invoice via Atomiq
2. **Privacy Phase**
   - Convert STRK → Lightning BTC (Atomiq)
   - Lightning BTC → Cashu e-cash (Mint)
   - Privacy mixing (shuffle, delay, split)
   - Cashu e-cash → Lightning BTC (Redeem)
3. **Output Phase**
   - Lightning BTC → STRK (Atomiq)
   - Transfer to Account B

- Update smart contract state

Risk Mitigation

Technical Risks

- **Integration Failures**: Comprehensive error handling and rollback
- **Timing Attacks**: Randomized delays and batch processing
- **Chain Reorganizations**: Confirmation requirements

Economic Risks

- **Exchange Rate Volatility**: Slippage protection mechanisms
- **Liquidity Issues**: Multiple liquidity sources and fallbacks
- **Fee Estimation**: Dynamic fee calculation

Required SDKs and Dependencies

Core Development SDKs

Starknet Development

```json
{
  "starknet.js": "^6.x.x",
  "cairo-lang": "^2.x.x",
  "starknet-devnet": "^0.x.x"
}
```

Bitcoin & Lightning

```json
{
  "ldk-node": "^0.x.x",
  "bolt11": "^1.x.x",
  "bitcoinjs-lib": "^6.x.x"
}
```

Cashu E-cash

```json
```

```
{
  "cashu-dev-kit": "^0.x.x",
  "@cashu/cashu-ts": "^1.x.x"
}
```

Cross-chain Integration

```
json

{
  "atomiq-sdk": "latest",
  "@braavos/wallet-api": "^1.x.x"
}
```

Frontend Development

```
json

{
  "react": "^18.x.x",
  "wagmi": "^2.x.x",
  "get-starknet": "^4.x.x",
  "tailwindcss": "^3.x.x"
}
```

Development Tools

```
json

{
  "scarb": "^2.x.x",
  "starkli": "^0.x.x",
  "cairo-test": "^2.x.x"
}
```

Specific SDK Capabilities

Atomiq SDK

- **STRK ↔ Lightning BTC swaps**: Proven integration with Braavos wallet enabling instant STRK payments at Lightning merchants (launched April 2025)
- **Cross-chain atomic swaps**: On-chain escrow mechanism with Bitcoin PoW consensus validation and minimal counterparty risk
- **Zero slippage**: Direct native Bitcoin to wrapped Bitcoin conversion via trustless protocol

Cashu Dev Kit

- **Lightning Integration**: Fund e-cash tokens via Lightning payments with full Lightning interoperability for instant withdraw
- **Privacy-First Design**: Anonymous bearer instruments ("nuts") with mint unable to track users, balances, or payment recipients
- **Minimal Core**: Lean library focusing only on core Cashu operations for optimal performance

Starknet.js

- **Account Management**: Multi-account support for privacy mixing
- **Contract Interaction**: Cairo smart contract integration
- **Transaction Management**: Batch operations and state tracking

Environment Setup Commands

```bash
# Install Cairo and Starknet tools
curl -L https://raw.githubusercontent.com/software-mansion/protostar/master/install.sh | bash
scarb --version

# Install Node.js dependencies
npm install starknet @cashu/cashu-ts bitcoinjs-lib

# Setup development network
starknet-devnet --host 127.0.0.1 --port 5050

# Install Python dependencies for Cairo
pip install cairo-lang starknet-py
```

Testing Infrastructure

- **Local Starknet**: starknet-devnet for contract testing
- **Lightning Testnet**: Regtest environment for Lightning operations
- **Cashu Test Mint**: Local mint instance for e-cash testing
- **Integration Tests**: End-to-end flow validation