Inheritor Beneficiary Claim Tool

User Manual

Table of Contents

Introduction

The Inheritor Beneficiary Claim Tool is designed to allow beneficiaries to claim digital inheritances that have become available to them. This tool provides direct access to retrieve, decrypt, and save inherited digital assets that have been encrypted and stored on the Arweave network.

With this tool, you can:

- Verify if an inheritance is in the "Claimable" state
- Retrieve the Arweave transaction ID from the blockchain
- Fetch and decrypt the symmetric key needed for asset decryption
- Download and decrypt the inherited digital asset
- Save the decrypted file to your local system

This tool is intended for use when an inheritance has reached the "Claimable" state, allowing you to claim and access the inherited asset securely.

Before You Begin

Before using this tool, you'll need:

1. **Recovery Mnemonic**: Your 12 or 24-word recovery phrase for the beneficiary account
2. **Gas Wallet Private Key**: A private key for a wallet containing ETH to pay for transaction fees
3. **Inheritance ID**: The specific ID of the inheritance you want to claim
4. **Network Information**: Knowledge of which network (Ethereum or Arbitrum) the inheritance is on
5. **Internet Connection**: Access to the Ethereum or Arbitrum networks

The inheritance must be in the "Claimable" state for successful retrieval. If you're unsure about the state of your inheritance, use the Inheritor Beneficiary Check Tool first.

Installation

Installing Node.js (Required)

This tool requires Node.js, which is NOT included by default in macOS, Windows, or Linux systems. You'll need to install it first:

macOS:

1. Option 1: Download the installer from Node.js website
2. Option 2: If you have Homebrew, run: `brew install node`

Windows:

1. Download and run the installer from Node.js website

Linux:

1. Ubuntu/Debian: `sudo apt update && sudo apt install nodejs npm`
2. Fedora: `sudo dnf install nodejs`
3. Arch: `sudo pacman -S nodejs npm`

Setting Up the Tool

Once Node.js is installed:

1. Save the script as `Beneficiary_Claim.js`
2. Open a terminal and navigate to the directory containing the script
3. Install required dependencies:

```
npm install ethers bip39 @ethersproject/hdnode axios crypto secp256k1
```

4. Make the script executable (macOS/Linux only):

```
chmod +x Beneficiary_Claim.js
```

5. To verify Node.js is installed correctly, run:

```
node --version
```

This should display the Node.js version (should be 16.0.0 or higher)

Running the Tool

To start the tool, run:

```
node Beneficiary_Claim.js
```

Initial Setup Process

1. **Enter Beneficiary Recovery Phrase**:
    - Type your complete mnemonic with all words separated by spaces
    - Example: `word1 word2 word3 ... word12`
    - Press Enter after entering all words
    - The script will display the derived address
2. **Enter Gas Wallet Private Key**:
    - Type or paste the private key of your gas wallet (with 0x prefix)
    - Example: `0x123abc...`
    - This wallet must contain ETH for transaction fees
    - The script will display the wallet address
3. **Select Network**:
    - Type either `ethereum` or `arbitrum` (case insensitive)
    - Press Enter to confirm
4. **RPC Configuration**:
    - Choose option `1` if you have your own RPC URL (Infura, Alchemy, etc.)
    - Choose option `2` to use public endpoints
    - If using option `1`, enter your complete RPC URL when prompted
5. **Enter Inheritance ID**:
    - Enter the complete inheritance ID starting with 0x
    - Example: `0x1234abcd...`
    - This must be a 32-byte hex string (66 characters including 0x prefix)

User Interface Tips

- **Inheritance ID Format**: Always enter the complete ID with 0x prefix
- **Waiting for Transactions**: Be patient during blockchain interactions
- **File Handling**: The decrypted file will be saved in the current directory
- **Confirmation**: Successful claim will show filename and file size

Understanding Each Function

The claim process involves several sequential steps that are performed automatically:

1. Check Inheritance Claimability

This function verifies that the inheritance is available to claim.

- **Process**:
    - Connects to the blockchain
    - Retrieves the inheritance details
    - Verifies the inheritance state is "Claimable"
    - Confirms you are the intended beneficiary
- **Important Notes**:
    - Only inheritances in the "Claimable" state can be processed
    - If the inheritance is in another state, the process will stop
    - No blockchain transaction is required for this check
- **When to Use**:
    - As the first step in the claim process
    - To verify an inheritance is ready to be claimed

2. Retrieve Arweave Transaction ID

This function gets the storage location of your encrypted asset.

- **Process**:
    - Queries the smart contract for the Arweave transaction ID
    - Converts the ID to the proper format for Arweave access
- **Important Notes**:
    - The transaction ID is stored on the blockchain
    - This ID is a pointer to where your encrypted asset is stored
    - This step doesn't require a blockchain transaction
- **When to Use**:
    - This is automatically performed after claimability is confirmed

3. Decrypt Symmetric Key

This function retrieves and decrypts the key needed to unlock your inheritance.

- **Process**:
    - Fetches the encrypted symmetric key from Cloudflare
    - Performs ECDH key exchange using your private key
    - Derives a shared secret for decryption
    - Decrypts the symmetric key
- **Important Notes**:
    - The symmetric key is encrypted specifically for your address
    - Only your private key can decrypt this symmetric key

    - This is a cryptographic operation, not a blockchain transaction

- **When to Use**:
    - This is automatically performed after retrieving the Arweave ID

4. Download and Decrypt Asset

This function retrieves and decrypts your inherited digital asset.

- **Process**:
    - Downloads the encrypted asset from Arweave network
    - Uses the symmetric key to decrypt the asset
    - Saves the decrypted file to the current directory

- **Important Notes**:
    - The filename is based on the inheritance ID
    - The file extension is determined from the asset metadata
    - This is the final step of the claim process

- **When to Use**:
    - This is automatically performed after the symmetric key is decrypted

Technical Background

Understanding Key Terms

Recovery Phrase (Mnemonic)

- A series of 12-24 words that generates your private key
- Must be entered with spaces between words
- Extremely sensitive information - never share with anyone
- In the Inheritor system, this recovers your beneficiary EOA address

Inheritance ID

- A unique 32-byte identifier for each inheritance
- Always begins with "0x" followed by 64 hexadecimal characters
- Used to track and access the inheritance on the blockchain

Symmetric Key

- Used to encrypt and decrypt the actual asset file
- The same key is needed for both encryption and decryption
- In Inheritor, this key is encrypted for the beneficiary's public key

Arweave Transaction ID

- A unique identifier for content stored on the Arweave network
- Points to where your encrypted inheritance is stored
- Converted between different formats during the claim process

EOA (Externally Owned Account)

- A standard Ethereum address controlled by a private key
- Derived from your recovery phrase
- Used to identify you as the beneficiary
- The script derives this from your mnemonic automatically

Cryptographic Process

The decryption process involves multiple cryptographic operations:

1. **ECDH Key Exchange**: Uses your private key and an ephemeral public key to create a shared secret
2. **HKDF Key Derivation**: Derives an encryption key from the shared secret
3. **AES-GCM Decryption**: Decrypts the symmetric key using authenticated encryption
4. **Asset Decryption**: The symmetric key is then used to decrypt the actual asset

This multi-layer approach ensures that only the intended beneficiary can access the inherited asset.

Inheritance States

An inheritance can exist in one of these states:

- **Designated** (Green): Not yet claimable, waiting for conditions to be met
- **Claimable** (Yellow/Orange): Ready to be claimed by the beneficiary
- **Claimed** (Blue): Already claimed by the beneficiary
- **Revoked** (Red): Cancelled by the testator
- **Purged** (Gray): Removed from the system

Only inheritances in the "Claimable" state can be processed with this tool.

Troubleshooting

Common Errors and Solutions

"Invalid mnemonic phrase"

- **Cause**: Recovery phrase words are incorrect or misspelled
- **Solution**: Double-check each word, ensure correct spacing, and try again

"Invalid private key for gas wallet"

- **Cause**: The private key format is incorrect
- **Solution**: Ensure the key includes the "0x" prefix and contains 64 hexadecimal characters after the prefix

"Connection failed"

- **Cause**: RPC provider is unavailable or rate-limited
- **Solution**: Try option 2 to use public endpoints, or use your own Infura/Alchemy key

"Invalid Inheritance ID format"

- **Cause**: The inheritance ID format is incorrect
- **Solution**: Ensure the ID is a 32-byte hex string with 0x prefix (66 characters total)

"Inheritance is not in Claimable state"

- **Cause**: The inheritance is not yet available to claim
- **Solution**: Use the Beneficiary Check Tool to verify the state and potentially trigger state changes

"Server did not return encrypted symmetric key"

- **Cause**: The CloudFlare service cannot find the key for this inheritance
- **Solution**: Verify you're using the correct inheritance ID and network

"Error during decryption"

- **Cause**: Issue with the cryptographic operations
- **Solution**: Verify you're using the correct beneficiary mnemonic

"Failed to retrieve asset from Arweave"

- **Cause**: Problem accessing the stored asset on Arweave
- **Solution**: Check your internet connection and try again

When to Seek Help

If you encounter persistent errors not covered above, please:

1. Take note of the exact error message
2. Do not share your mnemonic or private keys with anyone
3. Contact official support channels

Security Considerations

- **Use on a Secure Device**: Run the tool on a private, secure computer
- **Network Security**: Prefer a trusted network connection
- **Protect Your Mnemonic**: Never share your recovery phrase
- **Private Key Safety**: Your gas wallet private key is sensitive information
- **Temporary Use**: This is a specialized tool, not for regular use
- **File Storage**: Move claimed files to a secure location after decryption

Related Tools

The Inheritor Emergency Tools suite includes three complementary command-line applications:

1. **Testator Emergency Management Tool**: For testators to manage their Digital Will
2. **Beneficiary Check Tool**: For beneficiaries to monitor and check claimability of inheritances
3. **Beneficiary Claim Tool** (this tool): For beneficiaries to claim and decrypt inherited digital assets

If you're unsure whether your inheritance is claimable, use the Beneficiary Check Tool first before attempting to claim with this tool.

Disclaimer: This tool is provided for claiming inheritances only. While efforts have been made to ensure its security and accuracy, use it at your own risk. Always verify the effects of any blockchain transactions, as they cannot be reversed once confirmed.