Zero CLI: Zero-Knowledge Proof Command Line Interface

Overview

The Zero CLI provides a powerful, secure command-line interface for managing zero-knowledge proof identities, challenges, and verifications. Designed for developers and security professionals, it offers robust cryptographic operations with a simple, intuitive interface.

Prerequisites

- Node.js (version 16.0.0 or higher)
- npm (Node Package Manager)

Installation

Install the Zero CLI globally to use it from any directory:

```
npm install -g @obinexuscomputing/zero
```

Alternatively, you can install it in a specific project:

```
npm install @obinexuscomputing/zero
```

Identity Management Workflow

1. Creating an Identity

Create a secure identity from a JSON input file:

```
zero create -i identity.json -o id.zid
```

Example `identity.json`:

```
{
  "name": "John Doe",
  "email": "john.doe@example.com",
  "role": "Developer"
}
```

Command Options:

- `-i, --input <file>`: Input JSON file with identity data (required)
- `-o, --output <file>`: Output identity file (required)
- `-s, --salt <size>`: Salt length in bytes (default: 32)
- `-a, --algorithm <algo>`: Hash algorithm (sha256, sha384, sha512, default: sha512)
- `-f, --format <format>`: Output format (text, json, binary, default: text)
- `-v, --verbose`: Display detailed identity information

2. Verifying an Identity

Verify an existing identity against input data:

```
zero verify -i identity.json -k id.zid.key
```

Command Options:

- `-i, --input <file>`: Input JSON file to verify (required)
- `-k, --key <file>`: Key file for verification (required)
- `-d, --id <file>`: Optional separate ID file (if not embedded in key)
- `-v, --verbose`: Show detailed verification information

3. Deriving Specialized Identities

Create purpose-specific identities from a base identity:

```
zero derive -i id.zid -p "authentication" -o auth_identity.zid
```

Command Options:

- `-i, --input <file>`: Base identity file (required)
- `-p, --purpose <str>`: Purpose for derived identity (required)
- `-o, --output <file>`: Output derived identity file
- `-a, --algorithm <algo>`: KDF algorithm (default: pbkdf2-sha512)
- `-f, --format <format>`: Output format (text, json, binary)

4. Generating Challenges

Create a challenge for zero-knowledge proof verification:

```
zero challenge -o challenge.bin -s 64
```

Command Options:

- `-o, --output <file>`: Output challenge file (required)
- `-s, --size <size>`: Challenge size in bytes (default: 32)

5. Creating Proofs

Generate a zero-knowledge proof for an identity:

```
zero prove -i id.zid -c challenge.bin -o proof.bin
```

Command Options:

- `-i, --input <file>`: Identity file (required)
- `-c, --challenge <file>`: Challenge file (required)
- `-o, --output <file>`: Proof output file (required)
- `-f, --format <format>`: Output format (binary, base64)
- `-v, --verbose`: Display proof details

6. Verifying Proofs

Verify a zero-knowledge proof:

```
zero verify-proof -i proof.bin -c challenge.bin -d id.zid
```

Command Options:

- `-i, --input <file>`: Proof file (required)
- `-c, --challenge <file>`: Challenge file (required)
- `-d, --id <file>`: Identity file for verification (required)

System Information

View detailed information about the Zero library and CLI:

```
zero info
```

This command displays:

- CLI and library versions
- Protocol version
- Supported algorithms
- Current memory usage
- Active identities

Security Considerations

- Identities are cryptographically secure and cannot be reverse-engineered
- All operations use constant-time comparisons to prevent timing attacks

- Supports multiple hash algorithms with configurable parameters
- Implements secure memory handling to prevent data leakage

Troubleshooting

1. Ensure you have the latest version of Node.js
2. Verify that the input files are correctly formatted
3. Check file permissions
4. Use the `-v, --verbose` flag for detailed error information

Contributing

Contributions are welcome! Please visit our GitHub repository for more information: OBINexus Zero Library GitHub

Support

For additional support, please file an issue on our GitHub repository or contact support@obinexuscomputing.com.