ブロックチェーンの応用

SK3A 文家俊

研究背景

現在、セキュリティーにおいてデータ漏洩事件が多発されています。ブロックチェーンは高いセキュリティ特性を持つ技術であるため、

もし広く普及すれば、セキュリティがある程度強化できると思います。

目的

- ブロックチェーン技術で学生の作品に対してどのように利用できるかについて研修

研究内容

ブロックチェーンを学習し、分析して絞り込んで@1– 3

スマートコントラクトの脆弱性に集中

論文で資料を確保できるためにドメイン固有言語(DSL)という意見@ 4
DSLの例：HTML，CSS，SVG，Unity Script

現在であるNFTのコードがあるサイトに参考@ 5

Golangでブロックチェーンの基本コードを学習@ 6

@＝参考文献の番号

@5

Transfers | Holders | Inventory | Info | NFT Trades | Contract

Code | Read Contract as Proxy | Write Contract as Proxy | Read as Proxy | Write as Proxy

Search Source Code

Minimal Proxy Contract for 0x4a8ac7f22ded2cf923a51e4a1c67490bd8868add

Contract Name:	**TieredDrop**	Optimization Enabled:	**Yes** with **100** runs
Compiler Version	**v0.8.12+commit.f00d7308**	Other Settings:	**default** evmVersion

Contract Source Code (Solidity Standard Json-Input format)

Decompile Bytecode | Similar Contracts

File 1 of 54 : ERC721AUpgradeable.sol

```
// SPDX-License-Identifier: MIT
// ERC721A Contracts v3.3.0
// Creator: Chiru Labs

pragma solidity ^0.8.4;

////////// CHANGELOG: turn `approve` to virtual //////////

import "../../eip/interface/IERC721A.sol";
import "../../eip/interface/IERC721Receiver.sol";
import "../../lib/TWAddress.sol";
import "../../openzeppelin-presets/utils/Context.sol";
import "../../lib/TWStrings.sol";
import "../../eip/ERC165.sol";
import "../extension/Initializable.sol";

library ERC721AStorage {
    bytes32 public constant ERC721A_STORAGE_POSITION = keccak256("erc721.a.storage");

    struct Data {
        // The tokenId of the next token to be minted.
        uint256 _currentIndex;
        // The number of tokens burned.
        uint256 _burnCounter;
        // Token name
```

@6

```
❯ go run main.go add -block "first block"
2024/05/28 14:05:54 Replaying from value pointer: {Fid:0 Len:42 Offset:238}
2024/05/28 14:05:54 Iterating file id: 0
2024/05/28 14:05:54 Iteration took: 4.958µs
00002c3afdd8f7ee21f4dc366c79e3408da23f583847e19bbdc5a25b7e151560
Added Block!
```

```
❯ go run . print
2024/05/28 14:07:32 Replaying from value pointer: {Fid:0 Len:42 Offset:556}
2024/05/28 14:07:32 Iterating file id: 0
2024/05/28 14:07:32 Iteration took: 4.167µs
Prev. hash: 0000342dc11a9fd1833ed9fe18ca5627cedc56507de6698acfcafd301398cb35
Data: first block
Hash: 00002c3afdd8f7ee21f4dc366c79e3408da23f583847e19bbdc5a25b7e151560
PoW: true

Prev. hash:
Data: Genesis
Hash: 0000342dc11a9fd1833ed9fe18ca5627cedc56507de6698acfcafd301398cb35
PoW: true
```

今後の予定と課題

- Golangでブロックチェーンを作成
- DSL言語について合う言語を探す
- ポートフォリオにウォレットのようなボタンを設置し、
  過去の書類（例：成績書の写真）で実践

ご清聴ありがとうございます

参考文献

1. CoinEx アカデミー ｜ ノードとは何か、ブロックチェーン業界におけるその重要性を徹底解説
   https://www.coinex.com/ja/blog/2147-what-are-nodes
2. ブロックチェーン技術の歴史と展望
   https://cuc.repo.nii.ac.jp/records/6261
3. ブロックチェーン技術　ー学生の視点から現状と期待ー
   https://www.jstage.jst.go.jp/article/ieejjournal/137/10/137_708/_article/-char/ja/
4. Blockchain smart contracts formalization: Approaches and challenges to address vulnerabilities
   https://www.sciencedirect.com/science/article/abs/pii/S0167404818310927
5. Etherscan -NFT
   https://etherscan.io/nft-top-contracts
6. Building a Blockchain with Go
   https://www.youtube.com/playlist?list=PLpP5MQvVi4PGmNYGEsShrlvuE2B33xV1L