

Configuration Manual

LedgerLock

Forensic Evidence Verification Framework Application

MSc Research Project (Thesis)

Cybersecurity

Sajid Hussain Jatoi

Student ID: X23265736

School of Computing

National College of Ireland

Supervisor: Ross Spelman

National College of Ireland
MSc Project Submission Sheet
School of Computing

Student Name: SAJID HUSSAIN JATOI

Student ID: X23265736

Programme: MSc Cybersecurity **Year:** 2025

Module: **MSc (Research) Practicum/Internship (Thesis)**

Supervisor: ROSS SPELMAN

Submission Due Date: 11 DECEMBER 2025

Project Title: Configuration Manual

Page Count: 9

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Sajid Hussain Jatoi

Date: 11 December 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

1. Introduction

This configuration manual contains all the data and information of the system used, requirements in this research. This manual outlines the detail description of **data collection, code snippets of data preprocessing, model implementation and evaluations**.

DESKTOP-RM2Q4A6
HP EliteBook 840 G4

Rename this PC

Device specifications

Copy ^

Device name	DESKTOP-RM2Q4A6
Processor	Intel(R) Core(TM) i7-7600U CPU @ 2.80GHz 2.90 GHz
Installed RAM	32.0 GB (31.9 GB usable)
Device ID	6B581B07-575B-404E-8F4D-68A81B2D90A3
Product ID	00330-80000-00000-AA247
System type	64-bit operating system, x64-based processor
Pen and touch	No pen or touch input is available for this display

Windows specifications

Copy ^

Edition	Windows 11 Pro
Version	23H2
Installed on	14/05/2024
OS build	22631.6199
Experience	Windows Feature Experience Pack 1000.22700.1108.0

[Microsoft Services Agreement](#)
[Microsoft Software Licence Terms](#)

2. System Requirements

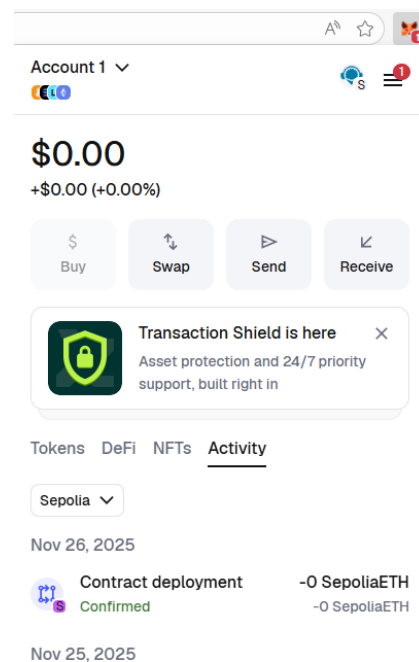
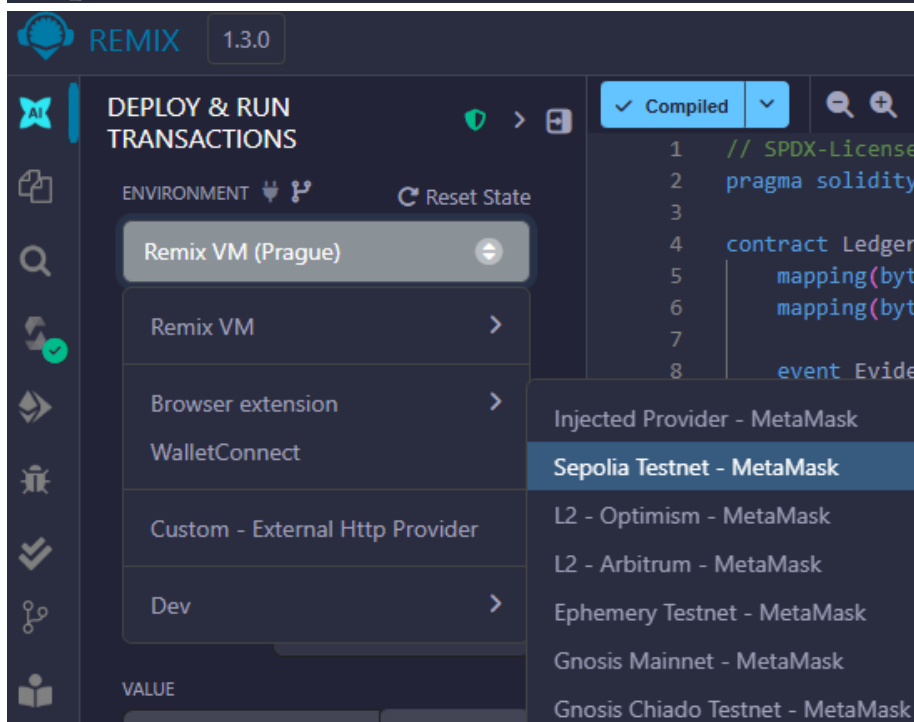
- **Operating System:** Windows 11
- **Programming Language:** Python 3.11 with dependencies (web3, fpdf, tkinter, pyinstaller, requests).
- **RAM:** Minimum 4 GB
- **Disk Space:** 500 MB free for logs, reports, and offline queue storage.
- **Network:** Internet connection required for online anchoring and verification; offline mode supported as well for differed synchronization
- **Blockchain Integration:** Smart contract deployment on Remix IDE, MetaMask wallet browser plugin and wallet
- **Sepolia testnet network:** Infura API key and address

3. Implementation

3.1 Smart Contract Deployment

Smart contract was deployed using Online Remix IDE with MetaMask Wallet using browser extension method

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 contract LedgerLock {
5     mapping(bytes32 => string) public evidence;
6     mapping(bytes32 => string) public txByHash;
7
8     event EvidenceRegistered(bytes32 indexed hash, string description, string tx_id);
9
10    function registerEvidence(bytes32 hash, string memory description, string memory tx_id) public { infinite gas
11        require(bytes(evidence[hash]).length == 0, "Evidence already exists");
12        evidence[hash] = description;
13        txByHash[hash] = tx_id;
14        emit EvidenceRegistered(hash, description, tx_id);
15    }
16
17    function getEvidence(bytes32 hash) public view returns (string memory) { infinite gas
18        return evidence[hash];
19    }
20
21    function getTxId(bytes32 hash) public view returns (string memory) { infinite gas
22        return txByHash[hash];
23    }
24 }
```



3.2 Anchoring Evidence to the blockchain

Application Binary Interface (ABI) interacting with the deployed smart contract and use as inputs metadata mapping using Web3.py load and call the below functions in snippets.

```
# -----Blockchain Setup-----

INFURA_URL = "https://sepolia.infura.io/v3/de22498fc7104273a9f6a65fc385a621"
CONTRACT_ADDRESS = "0xcel1c98c4B7A85Ca2B6d10cB19dB5D44d3080097E"
PRIVATE_KEY = "0x7b3d21685a9147dae57fcb996761b538271bee0d31e98dad29099bad6b210dde"
CHAIN_ID = 11155111

ABI = [
    {
        "inputs": [
            {"internalType": "bytes32", "name": "hash", "type": "bytes32"},
            {"internalType": "string", "name": "description", "type": "string"},
            {"internalType": "string", "name": "tx_id", "type": "string"}
        ],
        "name": "registerEvidence",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [{"internalType": "bytes32", "name": "hash", "type": "bytes32"}],
        "name": "getEvidence",
        "outputs": [{"internalType": "string", "name": "", "type": "string"}],
        "stateMutability": "view",
        "type": "function"
    }
]

# -----Initialize Web3 safely-----

try:
    w3 = Web3(Web3.HTTPProvider(INFURA_URL))
    account = w3.eth.account.from_key(PRIVATE_KEY)
    contract = w3.eth.contract(address=CONTRACT_ADDRESS, abi=ABI)
except Exception as e:
    messagebox.showerror("Blockchain Error", f"Web3 initialization failed: {e}")
w3, account, contract = None, None, None
```

- **CHAIN_ID** – a unique identifier assigned to each block used to prevent the replay attacks, ensure contracts are assigned for the correct chain block. This 11155111 is used by Sepolia tesnet network.

4. Application User Interface

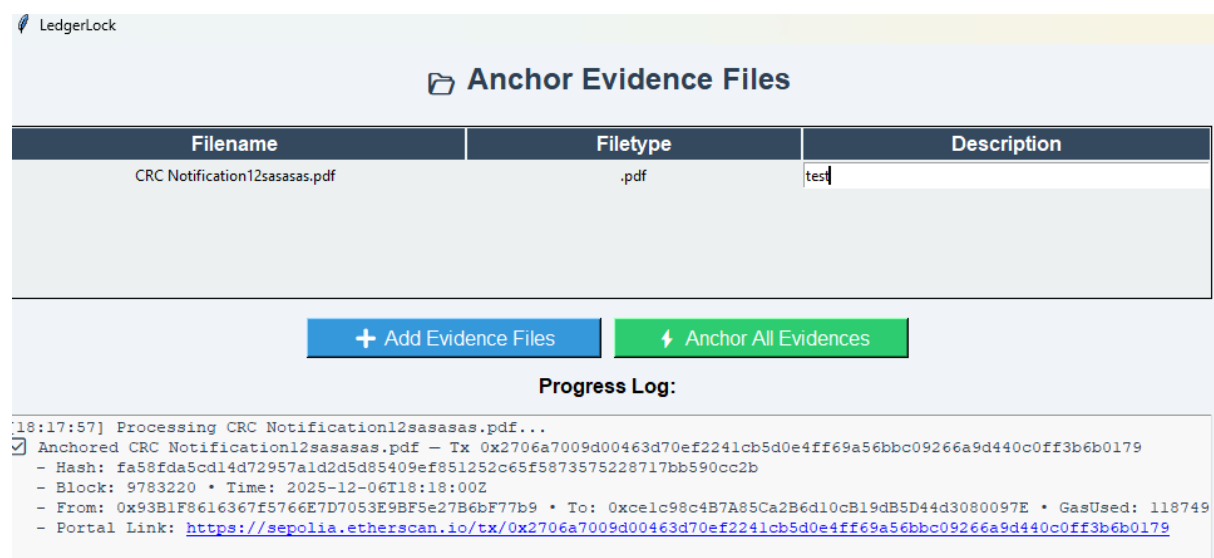
Launch LedgerLock application and use the functionality accordingly

4.1 Main Dashboard



4.2 Anchor Digital Records

Add Single / Batch of records – provide description against each record and once ready, click on the Anchor All Evidence. It will provide all interactive process logs in logs section for the user to update information real-time. It will first check if evidence exist already by checking the Keccak-256 hash code if doesn't it will provide all the meta-data and store this same for audit-trail on Sepolia Blockchain and export report.



If evidence already exist it will show an error and proceed for the next one, but it's designed to notify to the analyst in process logs and exported information as well.

Filename	Filetype	Description
CRC Notification12sasasas.pdf	.pdf	test
logo.png	.png	Logo

+ Add Evidence Files

⚡ Anchor All Evidences

Progress Log:

```
[20:43:38] Processing CRC Notification12sasasas.pdf...
X Failed CRC Notification12sasasas.pdf - Anchoring failed: Evidence already exists
[20:43:40] Processing logo.png...
[✓] Anchored logo.png - Tx 0x3bb4164fda75650fb35cac7eb312clead40048cfbd51f65d7be5c1a08a82588
  - Hash: 245f3da35d1e377f036b64112f827d0a0eale81706e4dc7d20a3f594118ab9e67
  - Block: 9783947 • Time: 2025-12-06T20:43:48Z
  - From: 0x93B1F8616367f576E7D053E9BF5e27B6bF77b9 • To: 0xcxcel98c4B7A85Ca2B6d10cB19dB5D44d3080097E • GasUsed: 118749
  - Portal Link: https://sepolia.etherscan.io/tx/0x3bb4164fda75650fb35cac7eb312clead40048cfbd51f65d7be5c1a08a82588
```

Record confirmation of Sepolia testnet blockchain

[illegible]

4.3 Verification of digital records

This functionality demonstrates that if the file and expected hash code matches than it proves the authenticity of the evidence and if it doesn't, than it will show that document hash either not related with the document or document is tampered.

Verify Evidence (single algorithm, per-file expectations)

+ Add File

⚡ Run Verification

🗑 Clear session

📄 Export CSV

📄 Export PDF

Algorithm

☒ Keccak-256

☐ SHA-256

☐ MD5

File: offline_queue.jsonlType: .jsonl

Attach file

Remove

MATCH

Keccak-256 expected: 38eb8686a96d1d8a4d6ca8f471de000b1a07d44ff8984e9dec62edcb57d09d

Verification results log:

20:57:34] Running verification (Keccak-256) on 1 file(s)...
[offline_queue.jsonl [Keccak-256] - MATCH

If doesn't match system will show the message as Mismatch

File: offline_queue.jsonlType: .jsonl

Attach file

Remove

MISMATCH

SHA-256 expected: 0xe542479ef295ea40e439621b0b25044e060ccfd95dc3f5db534d3cac9b92d37a

4.4 Work Offline

Application allows to compute hashes as per choice and provide to export the JSON or CSV to upload later to anchor the evidence.

Offline Queue (No Internet)

+ Add File

🗑 Clear Session

📄 Export Queue (JSONL)

📄 Export CSV Report

Algorithm:

☒ Keccak-256

☐ SHA-256

☐ MD5

File: offline_queue.jsonlType: .jsonl

HASHED

Remove

Data recorded in the JSON file - This contain exactly what is required to be uploaded and store information on the blockchain portal.

```
{
  "evidence_id": "81763731-9bd5-46b4-ae0b-e5a003bee800",
  "filename": "offline_queue.jsonl",
  "filetype": ".jsonl",
  "hash_algorithm": "Keccak-256",
  "hash_value": "38eb8686a96d1d8a4d6ca8f471de000b1a07d44ff8984e9dec62edcb57d09d",
  "collected_at": "2025-12-06T20:56:39.525448"
}
```


4.5 Upload Files

This will upload the same format file that was exported during the work offline. While anchoring the data to store on blockchain it will check if compute hash and evidence already exist or not if does, this will show that how many uploaded as below.

Online Anchoring (Upload Offline Queue)

Load Offline Queue (JSONL)

Anchor All Records

Reset

Back to Home

Export CSV

Export PDF

Progress Log:

```
/ Reset completed. Load a queue to start.
✓ Loaded 3 records from C:/Users/G4SAdmin/Documents/New folder/myfiles.jsonl
✗ Anchoring started.
[1/3] Submitted IMG_20240707_192037_1.jpg: 0x3362b79a43a807b80d50ffd57eb5d8b6883349a7aa22e952dbae3b5070c0e57e
✓ Anchored IMG_20240707_192037_1.jpg - 0x3362b79a43a807b80d50ffd57eb5d8b6883349a7aa22e952dbae3b5070c0e57e
✗ Failed IMG_20240707_192158_2.jpg - Hash length 62 is not 64 hex chars (bytes32 required).
✗ Failed IMG_20240707_192224_1.jpg - Hash length 32 is not 64 hex chars (bytes32 required).
✓ Finished anchoring. Success: 1 | Failed: 2 | Duration: 13.3s
```

4.6 Reports

LedgerLock application provide two options currently to export reports in cvs or pdf format. This is available in each section to export and show data accordingly.

Report contains headers such as (Filename, Description given while anchoring the evidence, File unique hash code Keccak-256, Blockchain Transaction hash, Block used, Timestamp, from which account To the address, Gas Used as Transaction fees, Portal Link – URL to click and verify that on the Etherscan Portal).

- CSV – Format

Filename	hashHex	txHash	blockNum	timestamp	gasUsed	from	to	portalLink	status	reason
1.pdf	8b92c07a	0x0b7a32	9783298	2025-12-01	96295	0x03B1F86	0xc0c1c98c4B7A85Ca2B6d10cB19dB5D44d3080097E	https://sepolia.etherscan.io/tx/0xb7a3233661a546f695c5693624d157bb4eb672cf5a7c2a747cd3a0e42e580	SUCCESS	
CRC Notification12sasasas.pdf	0x025253c	0xd425617	9783299	2025-12-01	141365	0x03B1F86	0xc0c1c98c4B7A85Ca2B6d10cB19dB5D44d3080097E	https://sepolia.etherscan.io/tx/0xd42561750dbcd234dc017ea8ab1d4bc0bccfdcb4fb7879412d3dd8be644	SUCCESS	
r.pdf									FAILED	Hash length 32 is not 64 hex chu

- PDF Format

LedgerLock

Distributed Ledger Technology (Blockchain) Digital Footprints Anchoring Report

Exported: 2025-12-02 21:28:18

Filename: r.pdf

Description: t

File Hash: c07bb6bfcb046de708126d393087dc9860e74e60cb218bf84b71c63ec40dda38

BlockChainTX Hash: 0x65c6b33cf0f8a88b51794becdded09f18cd84789eaa50e19f9a17a7871e6e0d8

Block: 9755605 | **Time:** 2025-12-02T21:28:12Z

From: 0x93B1F8616367f5766E7D7053E9BF5e27B6bF77b9

To: 0xc0c1c98c4B7A85Ca2B6d10cB19dB5D44d3080097E

GasUsed: 118713

Portal_Link:

<https://sepolia.etherscan.io/tx/0x65c6b33cf0f8a88b51794becdded09f18cd84789eaa50e19f9a17a7871e6e0d8>

8