

Scope Statement: Decentralized Test of Liveness



SUMMARY

1. Introduction
2. Image analysis
3. App architecture
4. Data Collection
5. Classifier Architecture
6. Local App Test
7. Cairo build
8. Cryptographic protection of labels
9. Final Tests and Updates
10. Bibliography

1/ Introduction

This project is to be realized by Petar Calic (me) under the supervision of Nicolas Bacca, VP of **Innovation lab at Ledger**, in the context of an internship of 6 months. I will be working mainly alone on the project with some assistance and guidance from Mr Bacca, the innovation team, the Data team, and many UPMC researchers that I will visit for consultations.

Know Your Customer (KYC) procedures are a critical function to assess customer risk and a legal requirement to comply with Anti-Money Laundering (AML) laws. Effective KYC involves knowing a customer's identity, their financial activities and the risk they pose.

Tests of liveness are a subgroup of tests designated to distinguish real and fake persons(bots). It consists of a set of technical features to counter biometric spoofing attacks where a replica imitates a person's biometrics or characteristics in order to deceive or bypass the identification and authentication steps given by the system. With the emergence of the internet with it, the daily increase of our everyday life dependence on its functionalities, increases the targets and vulnerabilities of third party malicious intentions. Here is one personal example: I made some nft's i wanted to give away freely to people that attended one online event i organized. The second I shared the link, a bot took all 100 of the tokens. A simple liveness check could prevent that. With AI capabilities to imitate humans rising every day, the task to discover bots that don't wish to be discovered, becomes more and more challenging.

However, there are still very successful tests like Google's CAPTCHA which relies on image recognition, mouse movements and a top secret server classifier. Our **goal** is to make an open source, completely decentralized Test of Liveness smart contract deployed on STARKNet, that would be accessible to anyone. There is also a more profound and general mission behind the project which is to make advances in the field of decentralized and open source technologies. We should start relying more on technologies that are verifiable than trustable.

An **important note** is that a test of liveness that relies purely on data collection and analysis, will never be 100% bot proof as the constant race of technologies between the two sides never stops. In order to be completely sure that we are dealing with a real person is to request a document verification for really critical systems. However we are confident that this test will repel more than 90% of most advanced bots. The cost of breach will hardly be profitable.

2/ Image analysis

- Minimum 500 images of different types, complicated with noise, rotations, color play etc...

3/ App development

Making tables of random 4x4 images from the database asking the user to click on 3 of the 16 pictures. The app collects data that are features that will be useful for the classification.

4/ Data Collection

Users that interact with the system prior during the conception phase will be used for building the dataset that will be used for classification training. The features need to be wisely chosen as we aren't going to be able to collect a lot of data and the information available is very restricted. We have to keep in mind that features that aren't impacting the classification are prone to make the learning process overfitting.

5/ Classifier Architecture

Choosing a simple but effective classifier. Make synthetic responses using a GAN. More cases need to be trained: an attacking ML that sees the answers, that partially sees, that is completely blind. Or maybe choosing a classifier that would make it as hard as possible exploiting the fact that the trained classifier is open source. For example using LSTM or Cascades or Haar.

6/ Local App Test

First test the classifier. Then if the results are satisfactory (more than 90% of successfully distinguished bots from humans). Then Testing the full app. Must recognize 95% of humans as humans.

7/ Cairo build

Learn Cairo and build the program to be deployed on Starknet.

8/ Cryptographic protection of labels

The system has a whole which can be exploited with medium effort if the attacker has knowledge in Machine Learning. Otherwise the test of liveness itself is very robust and we are confident that it is impossible to breach the test using individually made ML bots.

The attacker needs two elements in order to breach the system: Image labels (answers to the test), and the ML classifier. If retrieved the user can train it to find a set of features which satisfies the classifier and together with visible image labels the attacker can easily “act” as a human. If we find a way to protect one or those two elements the test will be almost 100% bot proof. See encryption methods, Time protection, ZK proofs...

9/ Final Tests and Updates

Test the smart contract.

10/ Bibliographie