Galactica's Zero Knowledge KYC Design

August 22, 2022

Version: 1.0

Abstract

Current cryptocurrency industry trends more often than not place AML/CTF regulations on the opposing side of the aisle under the guise that efforts towards KYC mechanism improvements detract from the trustless environments being developed. By and large the absence of such flexible solutions has created a significant blindspot for all cryptocurrencies. However, with the growing wealth of research behind zero-knowledge proofs, a solution with minimal compromises can be achieved. One that satisfies both the institutional demands on the AML/CTF side and consumer demands for privacy and security of personally identifiable information (PII). This concept; zkKYC, is the basic unit of provable-identity for what is termed Decentralized Society [1], which expresses identity attributes that can privately interface with dApps, smart contracts, and other Web3 entities. zkKYC [2] enables true development of reputation systems and meaningful social composability in Decentralized Autonomous Organizations and other Web3 governance systems. This innovation is vital to the preservation of user privacy and PII, while also ensuring trustless systems maintain their integrity from the ever-increasing threats of malicious actors. These social and governance systems can be deployed completely on-chain. This reaffirms the fact that crypto systems, while preserving transparency, can also adhere to compliance standards at the same level as those in TradFi.

Introduction

In the last few years decentralized applications (e.g. DeFi ecosystems, NFTs, GameFi) as well as their governance layers have gone through multiple stages of evolution. The mechanics underpinning the long-term growth of these dApps have been constantly being reformulated, adapted, and improved with the goal of facilitating adoption. However most developments have failed to address a more controversial but necessary component, which shares an outsized level of importance for securing institutional adoption - via on-chain Know Your Customer (KYC) [3].

The term KYC is generally mentioned in reference to trading on centralized exchanges [4], participating in investment rounds [5] and in more contentious cases, dealing with regulatory requirements. For many Americans or residents of jurisdictions [6] that are not so amiable to the concept of cryptocurrencies, KYC is something that is often dreaded. It means that certain users will likely be left out of a major part or all of the functionality of a platform. KYC does not have a positive relationship [7] with the majority of industry participants as many believe that the notion of KYC runs contrary to the very founding ideals of crypto.

The misplaced animosity towards KYC has an unintended by-product: a divorcing of much of the innovative energies that pervade other realms of cryptocurrencies so that KYC as a domain of technologies (in the context of its applications to crypto) hasn't evolved as much compared to its counterparts like

DeFi [8][9]. Performing KYC procedures hasn't changed much from the time of the ICO (to the chagrin of many smaller investors [10]) and while improving the technology underneath KYC may not immediately appear to benefit users like the outgrowths of DeFi managed to do, its development is monumentally important.

The Layer-2 and rollup technology race [11][12] that erupted on Ethereum has given way to a rejuvenation in other domains of knowledge being applied to cryptocurrencies. New emphasis was placed on privacy [13], crypto-focused applications of AI [14], and federated learning [15][16]. This additional effort gave an impetus to the rapid development of Zero-Knowledge proofs [17] and with this reapplication of thought and creative energies KYC has found a new crop of support [18].

By extension, zkKYC accounts can contain other data apart from that pertaining to the KYC process such as ownership of private Soulbound tokens (SBTs), educational YouTube channel, reputable Medium account, Summa Cum Laude designation of their Master's Degree, and et cetera. This is important as naturally, this data can be used in a multi-party computation setting and selectively proven or revealed.

This paper will go on to provide the technical basis of KYC's integration with Zero-Knowledge proofs and more importantly it will demonstrate how valuable zkKYC [2] technology can be in reputation-based, governance and general smart contract environments [19]. To unlock the next stage of secure, identity based user interaction and social composability on the road to a Decentralized Society, this paper will explain how and why zkKYC is that technical foundation.

Overview of the Technical Design

Schematic Overview

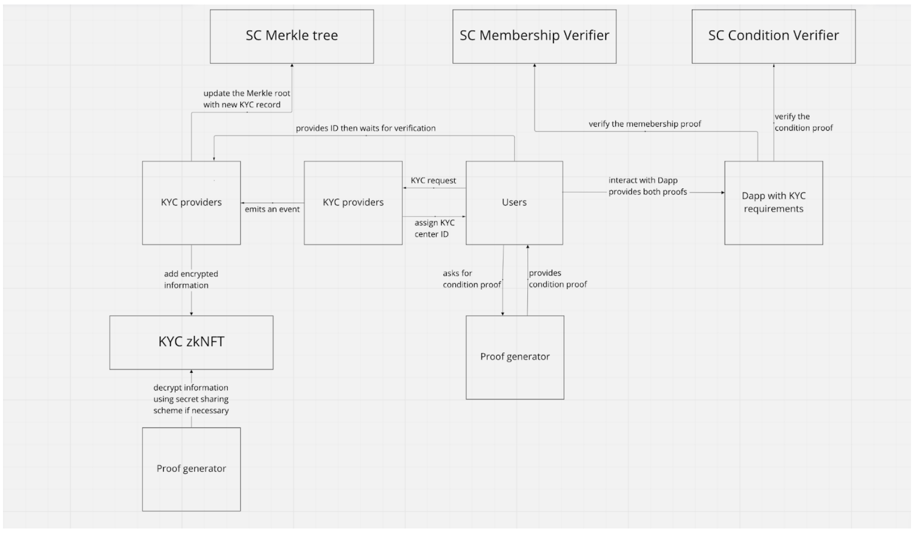
SC Merkle tree
SC Membership Verifier
SC Condition Verifier
update the Merkle root with new KYC record
provides ID then waits for verification
verify the memebership proof
verify the condition proof
KYC request
KYC providers
emits an event
KYC providers
assign KYC center ID
Users
interact with Dapp provides both proofs
Dapp with KYC requirements
add encrypted information
asks for condition proof
provides condition proof
KYC zkNFT
Proof generator
decrypt information using secret sharing scheme if necessary
Proof generator

Figure 1: Galactica's zkKYC Overview

KYC Record Creation

1. A hash is stored in a leaf of the Merkle tree: Each KYC user record will contain at least some of the following information: Public Key (wallet), Age, Country of Origin, Name, Verification Level, Random Salt and other.

2. The Merkle leaf can also be considered a "soulbound" [20] NFT - that is, it is a non-transferable NFT. Throughout the remainder of this text and in other documents we will refer to such tokens as SBT (soulbound tokens) [1].

3. At the moment, two methods of constructing the Merkle trees are investigated:

   a) Method 1: Only store the Merkle root on-chain and both the root transition and KYC record validity is verified by a ZK-proof;

   b) Method 2: Using Incremental Merkle tree [21], where alongside the Merkle root a filled subtree is also stored on-chain, which makes appending a new leaf independent of the current Merkle root possible. We still use a ZK-proof to verify the KYC record validity;

4. In this current version of the implementation we opt for the second method, as it allows more KYC providers to work on the same Merkle tree at the same time, whereas the first method might create concurrency issues;

5. Encrypted token: Alongside with the Merkle tree, the KYC provider will also post the encrypted KYC record on-chain. This information can be decrypted using k out of n governance keys if necessary;

6. In the Merkle tree smart contract a nullifier mapping will be used to record the valid hash of the KYC record, as the KYC record can be revoked, changed, and etc. In that case, the old KYC record is still in the Merkle tree, however it is marked as false in the nullifier mapping.

Membership Proof

1. This proof demonstrates that the user has a valid record in the Merkle tree;

2. The *public inputs* are the Merkle root stored on-chain and the user's address;

3. The *private inputs* are the Merkle path and the KYC record information that along with the user's address hashes to the corresponding Merkle leaf;

4. It has to be verified that the user's address is the one in the KYC record and that the Merkle path is valid - that is, it subsequently hashes towards the Merkle root stored on-chain;

5. It also has to be checked that this KYC record has not been discarded by checking the nullifier mapping.

Proof Generator and Condition Proof

1. The proof generator can be created from the public circom code, so it does not have to rely on one centralized identity;

2. The proof generation requires the user's private inputs, therefore it can be done for example on the frontend - the user's information remains private and secure;

3. Various condition verifies will exist such as Age thresholds, Country restrictions, and KYC level restrictions;

4. Anyone can deploy a new verifier depending on the condition that has to be checked.

Selective Information Disclosure

1. Users can publicly disclose any part of information and provide the proof with other information as private in order to demonstrate that the disclosed information is indeed the one contained in the hash stored on-chain. In this case users reveal this piece of information to everybody;

2. If a user only wants to reveal the information to a certain entity, then that user can encrypt the information with that entity's public key. The proof will be more complicated in this case.

Interaction with dApps

1. To interact with any protocol requiring KYC users will need to supply two proofs:
   a) The Membership Proof;
   b) The Condition Proof;
2. The protocol will then verify these proofs through respective verifiers and only proceed when the proofs pass.
3. As mentioned earlier, the protocols can deploy the verifiers themselves if they require custom conditions - they need to publish the circom code, so that anyone can create the prover.

Further Considerations

Generalization into zkAccount

1. In the discussion above we have mentioned information related to the KYC process, however it can be generalized to any metadata;
2. Any information can be encoded as a normal field like Country or Age;
3. The disclosure or encryption is the same as the process described earlier.

Secret Sharing Scheme

1. We propose a Secret Sharing scheme based on Pauwels (2021) [2].
2. The Merkle tree of Galactica acts as Verifiable Data Registry and is the storage used for:
   a) Public identifier DIDs (Decentralized Identifiers) - representing IDs of issuers, holders, verifiers, governments, and the relationships between them. For example, each time a holder registers at a verifier a new DID for this holder-verifier relation is created to make the zkKYC independent of other registrations
   b) Revocation lists - Cryptographic list of verifiable credentials revoked by the issuer - e.g. expired credentials;
   c) zkKYC tokens encrypting information about:
      - Issuers for certificate/KYC;
      - Holder-Verifier combination.

Government investigations (e.g. fraud) require decryption of the zkKYC token to:

- Find the issuer of the KYC and obtain the information;
- Verify that the fraud claim is valid - that is, the holder interacted with the verifier.

3. Following from the aforementioned, the zkKYC token is naturally encrypted. The decryption keys are shared between members of the DAO (further details are to be provided separately) with Shamir's Secret Sharing scheme [22] as follows:

   a) If there are m keys and a minimum of n are required to decrypt the secret, a polynomial of degree n is created with the secret being one coefficient and $n-1$ random coefficients. Every of the m participants receive a point on this polynomial. With n points, the polynomial can be reconstructed.

   b) Each point is encrypted with the government entity's public key on-chain.

Decryption of KYC Token

1. Process in case of government investigations:

   a) A government regulatory body approaches the DAO (more on the mechanics will be disclosed in a dedicated document);

   b) The DAO processes the request through voting;

   c) At least n members need to agree on the request to be able to decrypt the zkKYC token using Shamir's secret sharing scheme;

   d) The decrypted zkKYC token reveals the holder and issuer DID;

   e) The government regulatory body can then request the actual KYC data from the issuer;

   f) The issuer discloses the KYC data to the government regulatory body as long as the DAO vote is successful.

2. When DAO members change they need to pass their Shamir secret sharing data to the next member.

Conclusion

What has been outlined in the above discourse on zkKYC encapsulates a paradigm shift for the entire industry, an industry that has, from its inception, treated any form of KYC as antagonistic towards the ideals of cryptocurrency. This paper began the discussion with a high-level schematic overview of zkKYC's technical underpinnings to introduce readers to the components of KYC's technological evolution. The discussion transitioned into a step-by-step explanation of how a record is created and how it is then proved as valid within the ecosystem. The membership proof is the mathematical notation by which said record is proved within the Merkle-tree and thus proves the user as real and valid.

The proof generator was introduced alongside the notion that individuals and organizations will be able to deploy their own verifiers which check specific conditions existing in a user's record. Guarantees to user privacy were also included via the explanation of how users are able to selectively disclose information on their record. Additional information was provided on how users will interact with KYC smart contract dApps, particularly, how to supply proper information to the respective dApps. An excerpt on zkAccount generalization, the secret sharing scheme and the decryption of a KYC tokens was included to close out the discussion on zkKYC.

With rather minor modifications, the efforts outlined in this paper can be readily applied to a host of dApps and Web3 interactions. This results of these zkKYC developments being more rapidly deployable to production ready and live environments. Moreover, this paper also addressed the straightforwardness and customizability of the entire zkKYC procedure. KYC should not exist in an unapproachable state in which it is painted as a monolithic *enemy* to Web3. zkKYC has been elucidated in its entirety, it is ready to be integrated into Web3 and it is what is needed to provide the technical foundation for development of highly complex reputation-based, governance and social systems.

References

[1] V.Buterin, E.Glen Weyl, P.Ohlhaver (2022). Decentralized Society: Finding Web3's Soul

[2] P.Pauwels (2021). A solution concept for KYC without knowing your customer, leveraging self-sovereign identity and zero-knowledge proofs

[3] N.Jones (2021). Institutional crypto adoption – will it happen?

[4] L.Dayly (2021). What Is KYC, and Why Do Crypto Exchanges Require It?

[5] K.Peters (2021). Simple Agreement for Future Tokens (SAFT)

[6] C.Orji (2022). Bitcoin ban: These are the countries where crypto is restricted or illegal

[7] A.Lielacher (2022). What Is KYC in Crypto & Why Is it Controversial?

[8] D.Sian (2021). AML/KYC Culture Needs an Overhaul: How Technology Can Help You Adapt

[9] J.Courbe (2020).Financial Services Technology 2020 and Beyond: Embracing disruption

[10] P.Strickland, A.Veneziano. ICO Compliance: AML, KYC, SEC & FINRA – What You Need to Know Before You Launch an ICO

[11] E.Oosterbaan (2022). Ethereum's Rollups Aren't All Built the Same

[12] V.Buterin (2021). An Incomplete Guide to Rollups

[13] V.Buterin (2022). Some ways to use ZK-SNARKs for privacy

[14] IBM. Defining blockchain and AI

[15] M.R.Behera (2021). Federated Learning using Smart Contracts on Blockchains, based on Reward Driven Approach

[16] Z.Wang (2021). Blockchain-based Federated Learning: A Comprehensive Survey

[17] V.Buterin (2021). An approximate introduction to how zk-SNARKs are possible

[18] E.Lastname (2021). Dive Into Zero-Knowledge Proofs & the Future of Crypto

[19] V.Buterin (2022). The different types of ZK-EVMs

[20] V.Buterin (2022). Soulbound

[21] Incremental Merkle Tree

[22] A.Shamir (1979). How to share a Secret