Telegram IVM

by Galactica.com
September 24, 2024

Table of Contents

Concept

1. The identity stack developed by Galactica.com can function as the identity backbone of the Telegram mini-app ecosystem. When connected to Telegram Cloud and TON blockchain, it will enable secure identity management and verification using zero-knowledge proofs and fully homomorphic encryption.

2. Users can create Galactica.com ID accounts linked to their Telegram accounts, secured by biometric verification/passwords or other user-specified methods. KYC providers issue ZK Certificates, which enable users to generate cryptographic ZK Proofs that verify specific information, such as age or residency, without disclosing sensitive personal data. Human IDs, unique to each user, ensure privacy and are effective in mitigating Sybil attacks.

3. The idea can be easily generalized to include any data originating on/cross/off chain. Sophisticated methods for computing reputation functions over encrypted data can be supported using FHE libraries.

4. Galactica.com stack supports both off-chain and on-chain proof sharing, allowing (decentralized) apps to securely verify user credentials.

5. The system can be integrated with various blockchains and supports reusable proofs for identity verification across the Telegram ecosystem.

6. Currently the aforementioned stack is developed and production ready for web using MetaMask Snaps.

Entities and Glossary

Account

Telegram account registered with Galactica using its TMA. One account has one holding key.

Holding key

An EdDSA key that is used to sign ZK Proofs and hold ZK Certificates. It is derived from Entropy Hash, user's Passkey and Recovery Phrase, and can be restored using Shamir's Secret Sharing.

Entropy hash

A random number generated during the account creation. It is then stored in the Telegram Cloud.

Passkey

A biometric hash (Face ID or similar) or a password chosen by the user during the account creation.

Recovery Phrase

A secret phrase that can be used to regain access to the Account, allowing recovery.

Commitment hash

A hash derived from the user's Holding Key that is shared with KYC Provider during the process of ZK Certificate issuance. The hash is used by the Provider as a unique user id to link the KYC record to.

KYC Providers

Centralized parties that work as intermediaries between Galactica and the Users, processing their documents and submitting the information to the blockchain. Users may use any provider that supports their jurisdiction to get a KYC. Currently supporting Swissborg.com. Soon to support SumSub KYC. Others on-demand.

KYC ZK Certificate

This certificate is a result of KYC Provider's work. It consists of 2 parts:

1. **Public part** - a Merkle leaf of KYC data, published to the Merkle tree in the blockchain.
2. **Private part** - the KYC data itself, encrypted with the user's holder public key. It can be decrypted by Galactica's Passport TMA and then used by the app to issue ZK Proofs.

A certificate has a lifetime of 1 calendar year and should be prolonged by the user then to avoid expiration.

ZK Proof

A cryptographic proof of some statement that does not disclose any more information than the statement itself. Galactica uses zero-knowledge SNARK proofs based on the Circom library. The proof can be verified on-chain by a smart contract and off-chain using the Galactica SDK. In the context of a KYC ZK Certificate, the proof can for example state that the user has completed the KYC process, lives in a non-sanctioned country, and has an age above 18 years.

Each ZK proof has private and public inputs:

1. The private inputs are required to construct a proof and stay secret. For ZK KYC it includes the secret KYC data containing name, birthday, and residence.
2. The public inputs are disclosed with the proof. They can parameterize the statement, such as the age threshold, or contain information for verification, such as a Merkle root to check if the ZK certificate has been issued on-chain.

ZK Proofs can be published in two ways:

1. **Off-chain**. After generation, the proof may be transferred to "requesting TMA" or the Website directly (redirect).
2. **On-chain**. After generation, the proof may be submitted to the chain in a transaction to a smart contract. It can automatically verify if the proof is valid, unlock further interaction or save that the submitter has passed a valid proof in form of a soul-bound token (SBT). All the proofs of the

same type are being issued by the same SBT contract (which verifies proof validity upon mint), meaning that dApps will have to only check if the address is holding a certain SBT.
To post a proof, a transaction is being formed by Galactica TMA and then transferred to any Wallet, supporting TON (or any other supported chain) for signature.

ZK proofs have an expiration that is either set to 6 months from the generation time or match the parent ZK Certificate expiration time.
Galactica aims to have a set of default ZK Proofs, including:

1. Age proof - over 18, over 21
2. Non-sanctioned proof (not present in sanction lists)
3. Non-US proof
4. Non-sanctioned jurisdiction

These proofs will be reusable, which means users will not have to generate them again for every new app, at least until they are expired.

Human ID

A special hash acting as a unique identifier for the person using a TMA. It ensures sybil resistance, meaning every person can only get one humanID per TMA, no matter what telegram account they use. It is also TMA specific, meaning the user will get different humanIDs for different TMAs. This improves privacy because cross-referencing humanIDs gets hard without the user's consent.
Thus the humanID forms a basement for Galactica SSO (see Logging in with Galactica section).

Technically, the humanID is computed using the Poseidon hash of the following tuple:

1. User's Full Name
2. Date of Birth
3. Citizenship
4. TMA identifier or dapp address
5. A fixed salt registered per person

KYC providers verify that the user's information is correct and standardized. During the first KYC registration, the salt is fixed on-chain by the provider. This works by giving the provider a hash of the salt as commitment. Only the user knows the salt, which is the preimage of that commitment. Therefore, not even the KYC provider can guess a user's humanID. On following KYC registrations or renewals, the provider always checks that the salt commitment hash matches the previously registered one based on the combination of the user's full name, date of birth and citizenship.

System parts

Galactica TMA

User-facing application, supporting storage and management of:

1. Holding keys

2. ZK Certificates
3. ZK Proofs

Storage (Vault)

The app uses Telegram Cloud Storage for the storage of all entities. All the sensitive data is stored in encrypted form and can be decrypted with the user's Passkey.
Users are able to access and use their KYC from any device by only having their Telegram account and the Passkey/Recovery Phrase.

Limits

Cloud Storage allows storing up to 1024 key-value entities per user/app, where the value is up to 4096 symbols. This size is enough to store the whole KYC ZK Certificate or a proof, which means it's possible to store up to 1023 ZK Certificates or Proofs, plus the Entropy Hash.

Functionality

The app allows the issuance of:

1. Keys for holding (signing) ZK Certificates.
2. ZK Certificates, with supported KYC Providers. A special page lists them all.
3. ZK Proofs:
    1. **Default -** supported by Galactica itself, can be pre-generated without any request from 3rd parties.
    2. **Custom -** can be generated only by request from a 3rd party App/TMA, where the request contains a statement to be proven.

Galactica TMA is not a wallet and has no connection to any of the blockchains. However, ZK Proofs can be posted to TON or any other supported blockchain using integrations with, TON Space and other wallets via TON Connect.

Registry contracts

A Registry contract contains a Merkle tree that is used to publish ZK Certificates. It is crucial to have all the certificates registered in a public and persistent Merkle tree to maintain verifiability of all the ZK Proofs generated using them. The system can work with only one blockchain, or with a few of them, by just copying all the records everywhere.

Circuit Library

Galactica provides a library of zero-knowledge circuits for common use cases and components. It is built on top of the basic Circom library and can be used as building blocks for custom ZK proofs. This allows developers to parameterize, adjust, and extend access requirements and selective disclosures to their own use cases.

Provider SDK

This is a library containing methods to issue, update, and revoke ZK Certificates on-chain. Developed and maintained by Galactica (GO, js).

KYC Providers

These are 3rd party applications (Galactica partners) capable of accepting identity documents with Commitment hashes to issue ZK Certificates. They utilize Provider SDK to communicate with Galactica.

Scenarios

The journey starts at Galactica TMA, where the account is being created. All the entities related to Galactica are being operated by this application.

Galactica's account is connected to a Telegram account, and all the data, including keys and certificates, is backed up in Telegram Cloud in an encrypted manner. The User uses a special password to decrypt his/her data in case recovery or modification is needed.

Account Creation

On account creation, the application generates some entropy as salt, and the user selects a Passkey (may be Face ID or similar) or a password. Based on these 2 the holding key is generated and the user gets the Recovery Phrase.
A Holding Key is used to issue (derive) Commitment hash, that can be shared with KYC Guardians to obtain a KYC ZK Certificate. After getting a ZK Certificates, users can sign ZK Proofs with it.

Obtaining KYC

1. The User selects one of the KYC Providers from the list of available providers.
2. The User is redirected from Galactica TMA to the KYC Provider's Website/TMA with Commitment hash.
3. On the KYC Provider side, a user can do one of the following:
    1. Submit documents for verification;
    2. Log in with an existing KYC'ed account (if supported).
4. Then a payment has to be made for the processing of documents. The User pays to the KYC Provider directly with a supported currency.
5. The KYC Provider verifies the documents.
6. If needed, the KYC Provider issues a new ZK Certificate for a user, first the public part, (letting the user download it), and then the private part to the blockchain, signing it with the User's Commitment hash. On the upload of the private part, the KYC Provider pays a fee to Galactica, in gas (GNET) tokens.

The downloading of a Certificate can be done by either:

1. Scanning the QR code with Telegram App; or
2. Redirect back to Galactica TMA with the required parameters.

Once the ZK Certificate is added to the Galactica TMA, it can be used to generate ZK Proofs.

Gaining Trust levels

To prove legitimacy, users may *enforce* their KYC issued by one of the KYC Providers by passing KYC again with the 2nd and 3rd KYC Providers. After doing so, users are able to confirm with the ZK Proof that the personal data is the same across all of them. Every new KYC passed in such a way will increase the level of Trust of an Account.

Third parties are able to provide different levels of service based on the Trust level of an Account.

Generating ZK Proofs

Normally, a ZK Proof is requested by a 3rd party, may be another TMA or Web App.

1. The 3rd party shares the statement that has to be proven and a prover file to Galactica TMA. The requester should also specify, where the proof should be transferred after generation (off-chain), and/or how it should be posted to the blockchain.
2. Galactica TMA receives the request and prepares the proof. User sees what was proven and the data that was disclosed.
3. User approves transfer of data to the 3rd party and/or on-chain issuance.
4. If needed (on-chain posting), Galactica TMA prepares the transaction and allows users to sign it from any convenient wallet.
5. User gets redirected back to the 3rd party to continue.
6. Proofs that were posted on-chain may be reused by different dApps given one blockchain address is used and the requirements (statements) these apps seek are the same.

Logging in with Galactica

Third-party apps may request users to Log In with Galactica, which practically means sharing the Human ID.

1. The User gets redirected to Galactica TMA;
2. The User approves data sharing;
3. The User gets redirected back to the 3rd party app, and the 3rd party app receives the Human ID.

Account recovery

The account can be recovered if the user has access to the Telegram account and the Recovery Phrase. After the recovery, users can assign a new Passkey.

User data obfuscation

Given that only Commitment Hashes are shared with KYC Providers, it’s impossible for them to connect the personal data to any on-chain/off-chain activity until the Offenses Investigation process has been performed.

Offenses Investigation

A process, allowing authorities to get a person's KYC in case of misbehavior. There is a form for special requests from authorities that begins the process of Offenses Investigation. When a new case is submitted to the form, the process is initiated, and all authorized institutions receive its details. Institutions vote (multisig), and if the quorum is achieved, the requester gets a person's internal ID and KYC Provider name, which can be used to obtain the documents from a KYC Provider.