CRUBN

SSI-Verit

Atharv Singh Patlan
Punna Hitesh Kumar

Mentor : Somnath Banerjee

The Idea

SSI Verit aims to create a system to let social media users create a source of truth against their social media posts.

Motivation

01

Why create something like Verit?

Goals of SSI-Verit

02

What we aim to acheive using Verit

Components

03

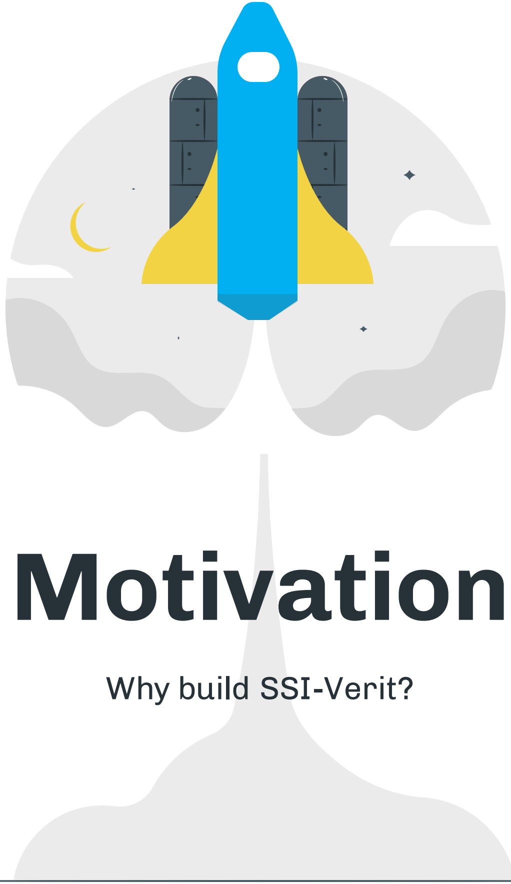
The different parts of the Verit system, and how they work together

Future Applications

04

Where all can Verit be used in the future, and how.

01



Motivation

Why build SSI-Verit?

Motivation

Untrusted Users

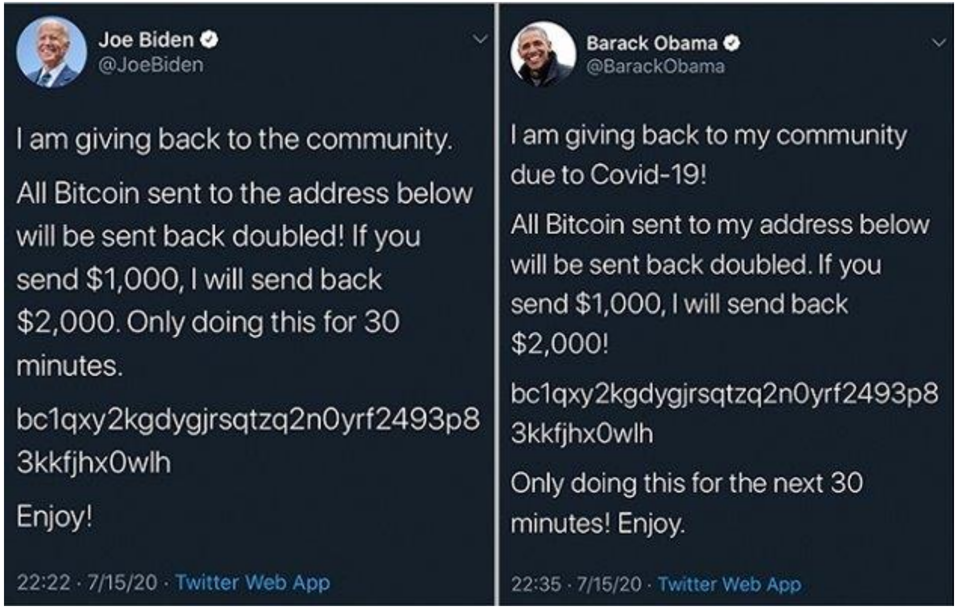
No way to prevent impersonation

Hacked accounts

No way to verify the actual user

Fake News

With senders taking no accountability



Joe Biden
@JoeBiden

I am giving back to the community.

All Bitcoin sent to the address below will be sent back doubled! If you send $1,000, I will send back $2,000. Only doing this for 30 minutes.

bc1qxy2kgdygjrsqtzq2n0yrf2493p83kkfjhx0wlh

Enjoy!

22:22 · 7/15/20 · Twitter Web App

Barack Obama
@BarackObama

I am giving back to my community due to Covid-19!

All Bitcoin sent to my address below will be sent back doubled. If you send $1,000, I will send back $2,000!

bc1qxy2kgdygjrsqtzq2n0yrf2493p83kkfjhx0wlh

Only doing this for the next 30 minutes! Enjoy.

22:35 · 7/15/20 · Twitter Web App

CRUBN

02

Goals

What we aim to achieve with Verit

Goals

Have only authentic users

Act as a profile verification system for Twitter

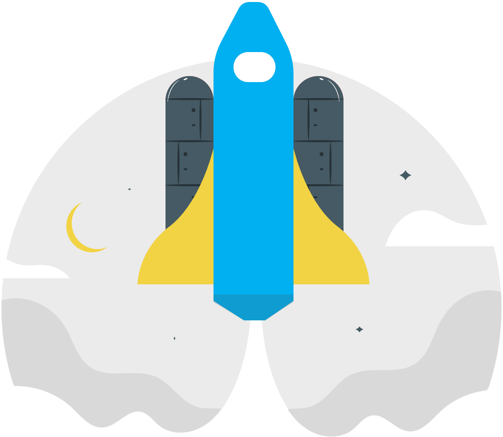
An additional layer of security

Extra defense to prevent impersonation.

Make users accountable

The signed proof of the post exists on the blockchain, for everyone to see.

03



Components

The what and the how of our idea.

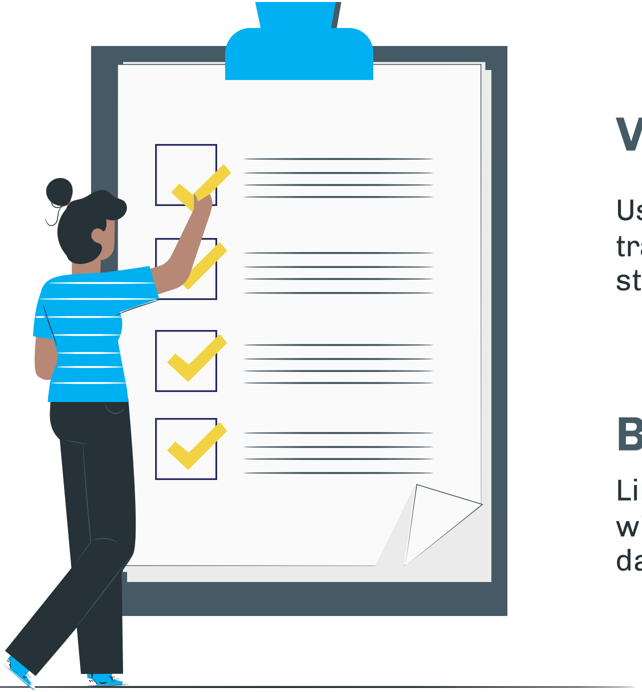
Components

Chrome Extension

User facing application end. Stores and manages accounts and sign tweets.

Smart Contracts

Deployed on a scalable blockchain, will act as verification and storage point for hashes and signs.



Verit Website

Used to verify tweets and transactions, and contains steps to sign up on Verit.

Backend

Links all the components with a set of APIs to retrieve data and call the contracts.

Chrome Extension

With a goal to facilitate interaction with Verit, it provides the following features:

Create and Manage Identity

Catch and Sign Posts

Smart Contracts

The Smart Contracts will act as the verification systems and data-stores, and will be responsible for verifying signatures and transactions before adding them to the blockchain.

There are 2 smart contracts that we have currently deployed:

1. **VeritIdentityTable**

2. **VeritTraceRecords**

VeritIdentityTable

VeritIdentityTable contract contains 2 data structures, User and Attestation, representing a user's address and social media handles corresponding to an address respectively.

It consists of 2 major functions:

- **registerAddress**(userType, veritSignature) - method to register the sender's address to the Verit ecosystem. The signature here is a signature on *hash*(‘Verit Platform Registration\n’ + senderAddress*)*, signed by Verit.

- **addAttestation** (address, Attestation) - method to add a new attestation to an existing address. An attestation will contain a user's platform handle, and will also contain a signature by Twitter on hash("Verit Platform Attestation\n" + userPlatformHandle + "\n" + senderAddress), to ensure that the handle is verified.

VeritTraceRecords

VeritTraceRecords contains a single data structure called Record, for storing tweet related data, such as the Tweet ID, the hash of the tweet message, and the user's signature to verify the tweet's correctness.

It consists of 2 major functions:

- **addRecord** (Record) – Records a new transaction containing message hash and signature of the (verified) user. The method recreates the inputMessage and verifies the signature against the public key stored in the IdentityTable. The inputMessage is constructed by hashing the original message text, with data such as tweet ID and platform handle, to uniquely assign a hash to a tweet.

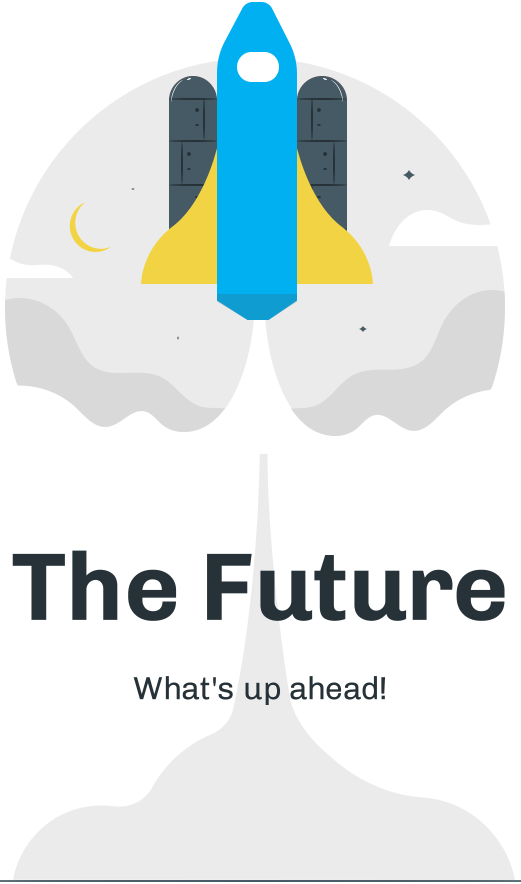
- **verifyRecord** (*index*, *hashOfOrigMessage*, *platformIdentifier*) – Can be used to verify a certain record, given the components.

Website & Backend

- The **website** will provide users a way to verify posts, and also register to SSI-Verit. Along with this it would provide features to view top verified profiles and posts.
- The overall **backend** will connect all these different components and provide APIs to call the contracts, verify posts, etc.

Currently, we have implemented SSI-Verit to work with Twitter

04



The Future

What's up ahead!

Job-Search portals

As many as

22 M

Fake profiles were detected by LinkedIn in 2019

Using Verit on Job-Search and Yellow Pages portals such as LinkedIn and Indeed can zero down the number of fake profiles.

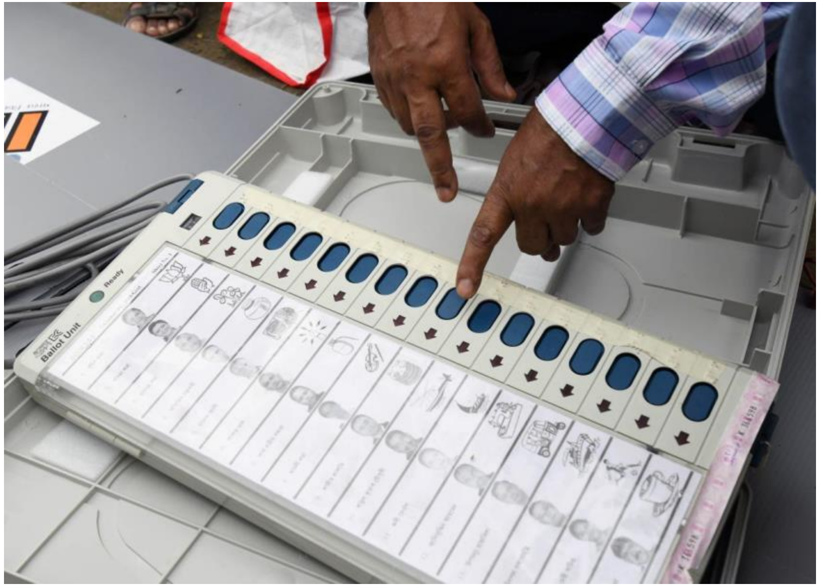
https://www.businessinsider.com/linkedin-releases-data-on-spam-scams-and-fake-account-removals-2019-11?IR=T

Posting Verifiable leads on P2P sale platforms

Verit can be used to verify leads about items on sale / purchase, and their proposed amounts on platforms such as EBay and OLX.

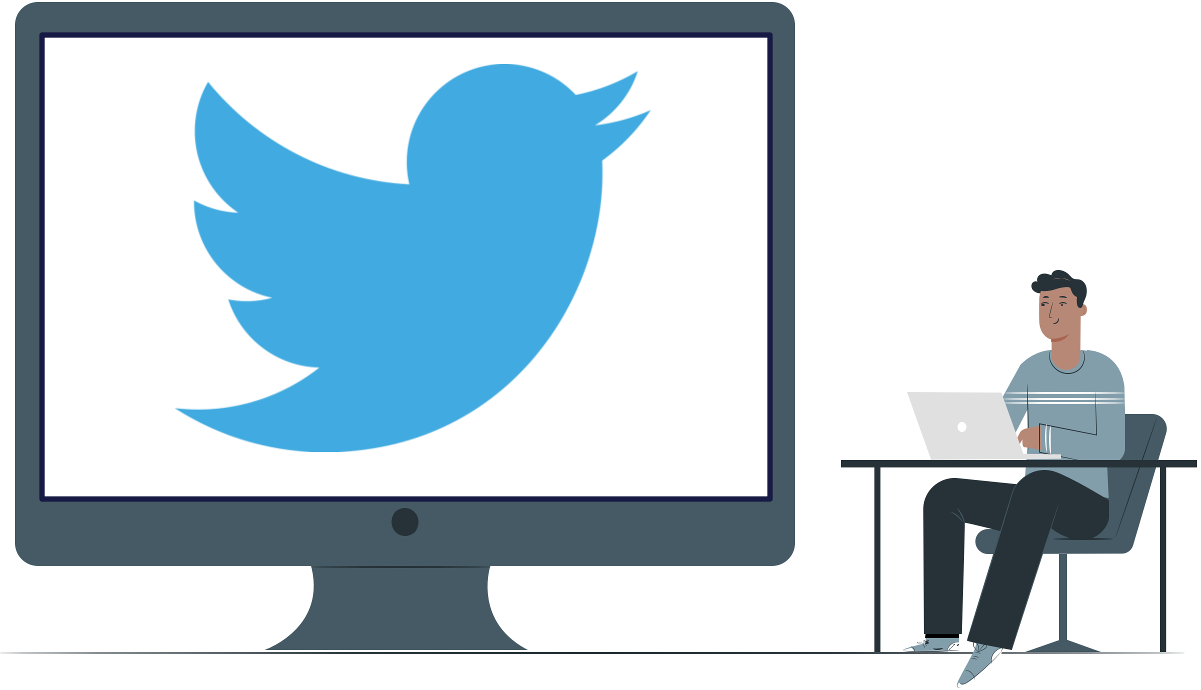
A step towards decentralized voting

Verit can combine with State authorities to verify user identities. Only identities signed by the govt. will be added to the blockchain.



Demo

Here is a short demonstration of our product:



Thanks!

Any questions?

The Idea

SSI Verit aims to create a system to let social media users create a source of truth against their social media posts. It is to serve as a source of message integrity and message authentication on social media platforms.

With Verit, we aim to increase the accountability of posts on social media, by storing their contents, in some form, immutably on the blockchain, with a signature of their sender. This would clearly establish the actual sender of the post, and save permanently the truth that the post actually contained the stored information.

Contents

So first I will be presenting as to why we decided to build something like Verit, and what goals it aims to achieve. Then we would move to the different working components of Verit, and finally to where we feel Verit can be used in the future.

Motivation

There has been a rampant spread of fake news and untrusted accounts on Twitter, with no way to prevent a person from creating an account impersonating someone else.

Cyber attacks are widespread, which hack into influencer's Twitter accounts, and disseminate fraudulent posts and schemes. As you can see in the image on the right side, the accounts of these influencers, or politicians, have been clearly hacked, however an ordinary used has no way to check if it has been a hack or is it genuine post. There is no way to verify if the sender from the account is the actual one or a hacker!

The third issue is the rampant spread of fake news. Even politicians do not stop to verify untrusted information, and share them away without any second thought about the consequences. If found to be fake, they simply delete / modify the tweet.

We felt the need to create a system to verify the sender of Tweets to be genuine, and if so, for the person to be accountable for the contents of their Tweets. Enter: SSI-Verit

Goals

Have only authentic users

Act as a profile verification system for Twitter, allowing only genuine users to be a part of the Verit ecosystem.

An additional layer of security

The Public Key Infrastructure adds an additional layer of defence, as in order to post a verifiable tweet from a hacked ID, the impersonator will have to gain access to the user's Verit private key

Make users accountable

Allow any user to verify the Tweets posted by a user, and check if the user is genuine or not.
As the user's signature is linked to the Tweet when stored on the blockchain, the user can be held accountable for the contents of their tweets, with the proof on Blockchain

Chrome Extension

The VERIT chrome extension is aimed at providing users easy access to the message signing functionalities and to interact with the overall Verit ecosystem. It has the following functionalities:

1. **Create and Manage Identity:** New identities can be created using the extension, and stored on the blockchain, with appropriate verification from the social media platform.

2. **Catching and signing posts:** We want the extension to automatically detect creation of new posts and facilitate it's signing by the user. Once signed, the hash of the message and corresponding sign are stored on the blockchain, to allow anyone to check that whether the sender is willing to sign and put their name on the post or not, and if the message text was originally what it says it is.

Smart Contracts

The Smart Contracts will act as the verification systems and data-stores, and will be responsible for verifying signatures and transactions before adding them to the blockchain.

There are 2 smart contracts that we have currently deployed:

1. **VeritIdentityTable**

2. **VeritTraceRecords**

VeritIdentityTable

Responsible for storing and verifying registered address, and link accounts at various social media platforms to an address. An account from a platform can only be linked when the platform allows it, by signing a message.

VeritIdentityTable contract contains 2 data structures, User and Attestation, representing a user's address and social media handles corresponding to an address respectively.

It consists of 2 major functions:

- **registerAddress**(userType, veritSignature) - method to register the sender's address to the Verit ecosystem. The signature here is a signature on *hash(*‘Verit Platform Registration\n’ + senderAddress*)*, signed by Verit.

- **addAttestation** (address, Attestation) - method to add a new attestation to an existing address. An attestation will contain a user's platform handle, and will also contain a signature by Twitter on hash("Verit Platform Attestation\n" + userPlatformHandle + "\n" + senderAddress), to ensure that the handle is verified.

VeritTraceRecords

Responsible for adding posts signed by the registered sender, to the blockchain. The hash of the post text is signed, and not the plain text, to maintain privacy. Thus, anyone with the original message and sender can verify the correctness of the displayed message, by checking if it's signed or not.

VeritTraceRecords contains a single data structure called Re, for storing tweet related data, such as the Tweet ID, the hash of the tweet message, and the user's signature to verify the tweet's correctness.

It consists of 2 major functions:

- **addRecord** (Record) – Records a new transaction containing message hash and signature of the (verified) user. The method recreates the inputMessage and verifies the signature against the public key stored in the IdentityTable. The inputMessage is constructed by hashing the original message text, with data such as tweet ID and platform handle, to uniquely assign a hash to a tweet.

- **verifyRecord** (*index*, *hashOfOrigMessage*, *platformIdentifier*) – Can be used to verify a certain record, given the components.

Website & Backend

- The **website** will provide users a way to verify posts, and also register to SSI-Verit. Along with this it would provide features to view top verified profiles and posts.
- The overall **backend** will connect all these different components and provide APIs to call the contracts, verify posts, etc.

Currently, we have implemented SSI-Verit to work with Twitter

Job-Search portals

So back in 2019, a really large number of fake profiles were detected on LinkedIn. If you think about it, it isn't very difficult to scam people on LinkedIn. It doesn't verify in any way if you are a part of a company, and anyone can pretend to be an employee of a reputed company to dupe people of their time and money.

Using verit companies can provide authorizations to people to add their names to their profiles, so that there is an evidence of the actual organization to which the user belongs

https://www.businessinsider.com/linkedin-releases-data-on-spam-scams-and-fake-account-removals-2019-11?IR=T

Posting Verifiable leads on P2P sale platforms

Verit can be used to verify leads about items on sale / purchase, and their proposed amounts on platforms such as EBay and OLX.

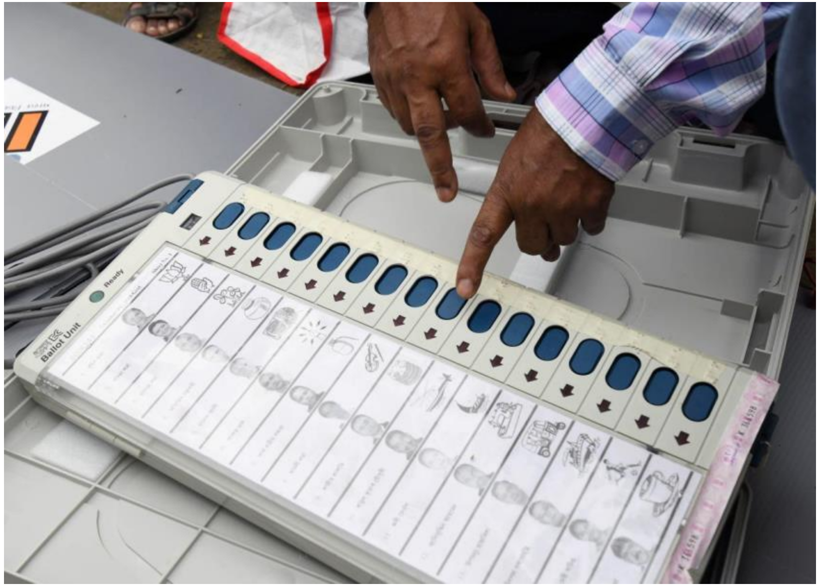
This will prevent cases in which posters abruptly increase the selling price, seeing high demand.

It can also be used to verify the identity of bidders for these goods.

A step towards decentralized voting

Verit can combine with State authorities to verify user identities. Only identities signed by the govt. will be added to the blockchain.

Using such verification methods will allow a verified voter to securely cast their vote, with the help of blockchains. A user can verify that their vote has gone only to the intended party.



Smart Contracts

The Smart Contracts will act as the verification systems and data-stores, and will be responsible for verifying signatures and transactions before adding them to the blockchain. There will essentially be 2 smart contracts:

1. **VeritIdentityTable:** Responsible for storing and verifying registered address, and link accounts at various social media platforms to an address. An account from a platform can only be linked when the platform allows it, by signing a message.

2. **VeritTraceRecords:** Responsible for adding posts signed by the registered sender, to the blockchain. The hash of the post text is signed, and not the plain text, to maintain privacy. Thus, anyone with the original message and sender can verify the correctness of the displayed message, by checking if it's signed or not.