**Decentralized Secure Collaborations (DSC): Litepaper**

**…**

**Abstract:** Todo

**Section 1: Introduction**

Todo

**Section 2: SCAMS on FINANCIAL TRANSACTION BETWEEN 2 or more PARTIES**

As mentioned above, DSC can be an efficient solution to frauds and scams in cases where financial transactions are involved between 2 parties/ among 3 or more parties. In this section, we study the numbers of recent such scams that have been committed in different sectors.

Firstly, let us study the scams involved in social media. Social media is undoubtedly a big thing today and our lives are surrounded by them or their effects. Different social media giants like Facebook, Instagram, WhatsApp, Twitter, Snapchat, even YouTube, etc. are some of the most discussed, most viewed, most studied items in today’s world. There are many financial transactions also involved in these social media. Let us take Instagram for example. This social media giant started its journey as a benign photos/videos sharing app, but quickly (probably since its acquisition by Facebook, now Meta) it started to create businesses among its users. There are many Instagram influencers now with thousands or millions of followers. They are probably the best way right now to reach a big ocean of audience. This can range from a professional Hollywood actor ‘paid-partnershipping’ with a well known business, to a not so well-known influencer promoting some local brand. Omnicoreagency.com states that there are [500,000 active influencers](https://www.emarketer.com/content/is-everyone-on-instagram-an-influencer) on Instagram.  [37% of Instagram users](https://takumi.com/research) interact with influencers, which equals to about 74million [1]. Now, let us study the frauds involved in this domain.

Chart, bar chart

Description automatically generated

Image courtesy: Statista [2]

The image above shows the percentage of influencers in Instagram involved with frauds. The bar charts are for 3 years- 2019, 2020 and 2021. The statistics also show different influencers according to their numbers of followers. We can see, on average 67% of influencers over 1 million followers are involved with frauds for these 3 years. It deserves mention that, many of the frauds are not even reported for different reasons like blackmail, poor customer support, lack of knowledge of users on how to defend him/her/themselves.

Most common types of frauds in social media are:

1. Phishing Scams,
2. Fake Investments,
3. Fake Collaboration Requests,
4. Selling Followers and Likes,
5. Giveaways,
6. Non-Existent Job Offers,
7. Imposter Brand Account, etc.

Most people who use social media have been affected with at least one of them, or came across at least one of them, or at least heard of such a scam.

The biggest social media giant, Facebook, reports removal of billions of fake accounts each year. It removed 1.6 billion fake accounts in Q1 of 2022 and 1.4 billion in Q2 of the same year. Fake accounts represented 5% of Facebook's worldwide monthly active users during Q2 of 2022, as Meta [reported](https://transparency.fb.com/data/community-standards-enforcement/fake-accounts/facebook/) [3]. No wonder, many of these accounts are involved in crimes, many of which involve monetary transactions.

Another social media giant that almost everyone from professional domains use everyday is LinkedIn. According to LinkedIn Transparency [4, 5], between July and December 2021, it stopped 11.9 million fake accounts from being created during the registration process, proactively restricted 4.4 million fake accounts, and restricted 127,000 accounts after members reported fake accounts. It also proactively removed 70.8 million accounts for spamming or scamming while 179,000 were removed following member reports of spam or scams. It must be mentioned that, most of these accounts advertise for non-existent jobs, or they claim to be agents connected to big companies, or claim they can write a perfect application, etc in return for money. One of the biggest threats here is impersonation, which also is involved in the frauds in Instagram mentioned above [6].

In the USA, more than 95,000 people or Consumers in 2021 reported losing about $770 million to fraud initiated on social media, according to the Federal Trade Commission's Consumer Protection Data Spotlight. That's about 25% of all reported fraud losses for 2021 -- an 18-fold increase from 2017, the FTC said [7].

According to CBS News [8], at least 15% of advertisers' spending on influencer marketing is lost to fraud, costing them $1.3 billion annually in 2020, which is supposed to reach almost $2 billion in 2022.

With the rise of cryptos and NFTs, these scams have increased exponentially. This is because promotions of these NFTs and new cryptos largely depend on collaborations with the influencers who have millions of followers in different social media. We would like to take some space to share our personal experience. Our own NFT’s promotion was done in collaboration with a team of influencers from Argentina from March 2022 to August of 2022. We had paid them a load of money for that. Unfortunately, almost no promotion was done from their side, they bought both followers for our handles at cheap costs, and others. To be honest, the biggest influence behind our work is this scam that we faced.

One big side of online fraud is counterfeiting, which means when someone sells, promotes, etc. some stuff, brand, etc. with the name of a famous brand, personality, etc. The sold item can be a physical item with tangible effects, or it can be something intangible, for example, promotions.

“The amount of total counterfeiting globally has reached to 1.2 trillion USD in 2017 and is bound to reach 1.82 trillion USD by the year 2020 which includes counterfeiting of all equipment/products from defense equipment to counterfeiting of watches.” The 2018 Global Brand Counterfeiting Report [9] estimates that the losses suffered due to online counterfeiting globally has amounted to 323 billion USD in the year 2017. According to the analysis made in the report, losses incurred by Luxury Brands because of sale of counterfeiting through the internet accounted for 30.3 billion USD.

Of course, another big market is e-commerce. According to statista, Retail ecommerce sales are set to reach $5.5 trillion in 2022 [10]. There are about 8 billion people worldwide of which about 2.14 billion people shop online [11].

### According to Payments Dive, eCommerce fraud caused losses of $20 billion in 2021 [12].

Chart, bar chart

Description automatically generated

Image courtesy: statista [13]

According to statista, $41 billion globally was lost on e-commerce payment fraud in 2022 [13].

In ecommerce, when a chargeback happens, it becomes cumbersome for the retailer themselves as they have to prove that the product was successfully delivered. Many customers take unfair advantage of this situation. They apply for chargeback even after receiving the product. This is also called friendly fraud. According to Verifi, businesses lose $308 for a $100 chargeback [14]. According to LexisNexis [15], this amount is almost $400. According to Cybersource [16], this is the top fraud affecting the merchants in all domains. According to Chargeback911 [17], the number of internet scam complaints is growing rapidly as shown in the figure below:

Chart, bar chart, waterfall chart

Description automatically generated

Image Courtesy: Chargeback911 [18]

According to Chargeback911, by 2023 6 out of 10 chargebacks in North America will be fraudulent [18]. According to Forbes [19],, chargebacks can account for between 40% and 80% of all eCommerce fraud losses. Mastercard [20] states that 80% of chargebacks are fraud-related. This includes both third-party (“criminal”) fraud, as well as first-party (“friendly”) fraud.

However, we do need chargeback as the ‘last resort’ of the customers against credit card frauds.

A new study from [Juniper Research](https://www.juniperresearch.com/home) [21] has found that the total cost of eCommerce fraud to merchants will exceed $48 billion globally in 2023, from just over $41 billion in 2022. According to the Nilson Report [22], by 2030, total payment card volume is expected to top $79 billion. Of that amount, an estimated $49.32 billion will be lost to criminal fraud.

Finally, fraud costs the global economy $5.127 trillion each year [23].

**Section 3: EXISTING APPLICATIONS**

Any business deal between 2 untrusted parties can be moderated by a third party that the other parties mutually trust. This can be defined by Escrow protocol. Escrow is a contractual agreement between two parties in which a trusted third party receives and holds the asset on behalf of the sender. The third party is usually called the escrow agent. The agent monitors the deal and either disburses the asset to the receiving party (in case the deal is complete) or returns the asset to the sender party (in case the deal is not fulfilled). The agent must follow some specific conditions set by the main 2 parties. The conditions are set prior to creating an escrow account. Not necessarily all the conditions must be satisfied in order for the deal to go through. There can be scenarios where if a majority of the conditions (say 2 out of 3) are satisfied then the agent can decide that the deal is complete. When the 2 main parties decide to create an escrow account, they agree on a set of conditions (can be provided by either of the parties), and the sender party deposits the asset to the account. Both parties agree on an escrow agent (can be a bank or a single entity). The agent remains responsible for the asset until a specific time period. If the conditions are satisfied within the agreed time period then the agent disburses the asset to the receiver party. If some of the conditions fail to satisfy or if the time period ends before the objective is completed then the agent returns the asset to the sender party.

Diagram

Description automatically generated

Up next we will discuss some common examples that require the use of escrow protocol.

*A. Real Estate Business*

A very popular use case application for escrow protocols is real estate transactions. In the real estate business, a trusted party is hired to hold all documents and assets on behalf of the buyers and sellers. The buyers are usually individuals or a collective entity and the sellers are property providers. The third parties are generally law firms or escrow companies. In the real estate business, when there is an agreement between both parties, there is often a down payment involved. The down payment is usually a part of the total cost of the property which the buyer must provide to the seller as an initial investment. The parties agree on a set of terms such as the number of installments for the rest of the payment, and legal documents from both parties. If an escrow protocol is involved, then this investment amount along with the conditions and documents are held by the third-party escrow agent. In case the buyer backs out or the agent finds any loophole in the documents provided by the buyer, then the initial investment amount is sent to the seller as compensation. On the other hand, if the seller backs out or becomes unable to provide the property within the agreed time period then the amount is returned to the buyer. During the initial agreement, the seller can also add some compensation fees that the buyer gets in case the seller backs out. In case multiple installments are involved, the buyer can send these payments to the agent as well. The agent, keeping a separate account for the next payments, sends only the down payment to the seller and the rest of the amount to the buyer in case the buyer backs out. In case the buyer sends the whole amount and the seller provides the property on time, the agent sends the whole amount (from both of the accounts) to the seller.

*B. Import-Export Business*

When 2 untrusted parties are involved in trading it is often a custom to have a third-party moderator. Either of the main parties can be a private company or the Government. An escrow protocol is needed in this use case because it can eliminate the possibility of the buyer having a damaged asset (or not having the asset at all) after advanced payment, as well as the risk of the seller not having the full payment after the asset is shipped. Once both parties settle on the trading cost, volume, and time of delivery, the buyer party sends the amount to the escrow agent. After receiving the asset from the seller party, the buyer party can check the condition of the asset and send a ‘green light’ to the agent. The agent, upon receiving the validation from the buyer, sends the amount to the seller party.

The escrow protocol is also applicable to e-commerce applications and other online marketplaces. In this scenario, after ordering a specific asset from an e-commerce application, the buyer sends the amount along with other necessary documents to the escrow agent. Upon delivery, the buyer uses a predetermined amount of time to inspect and accept the asset. Once accepted, the funds are released by the escrow agent to the e-commerce application. In case of a dispute (damaged asset or not receiving the asset within a specific time period), the agent can investigate the matter. Since the buyer themselves can damage the asset or deny receiving the asset to avoid sending the amount, one solution for the agent to properly resolve the dispute can be to use a supply chain management scheme for the overall shipment of the asset. This solution is very effective but it needs further involvement to implement the supply chain in trading. In many cases, this solution also needs multiple escrow agents to properly validate the shipment. Perhaps a simpler solution can be for the seller to ship the asset to the escrow agent, who in terms, can validate the quality of the asset and finally send it to the buyer.

*C. Crowdfunding*

It is quite common to use escrow protocol in crowdfunding. A crowdfunding campaign is involved when typically a startup initiates a project that requires funding. The main goal is to invite investors to donate funds in exchange for specific privileges such as stakes. The project coordinators need to provide details of the project along with a ‘target amount’ of funds that is required to fuel the project. In different crowdfunding applications, it is a mandatory criterion to mention the target amount and the time period of the campaign. In case the target amount is not reached within the predefined period of time, it is considered that the project does not have the necessary funding to be started. An escrow agent is required in this scenario who can return the existing funds to its actual investors in case the target amount is not reached. The general concept is, the escrow agent will hold all the funds (along with the details of the investors) until the end of the campaign. After the campaign time period is over, if the agent finds out that the target amount is reached, then he can send the whole amount to the project team. Otherwise, the agent will conclude that the campaign has failed to get the necessary amount and will return the funds to the investors.

Up next we will discuss some of the most common escrow applications.

*A. Centralized Applications*

**1. Escrow.com:**

It provides a centralized escrow service that allows users to buy and sell assets through a trusted third-party moderator (broker). The application’s three-party transaction gives Brokers the ability to start, manage, and close every sale privately. Escrow brokers provide a secure and neutral third-party service to facilitate transactions between two parties. They ensure that the terms of the agreement are met before releasing the funds or property to the appropriate party. All the 3 parties need to register to escrow.com in order to agree to terms. Once the parties agree to terms, the broker initiates the transactions. Next, the buyer pays the amount to escrow.com. Upon receiving the payment, escrow notifies both the seller and broker. In the next stage, the seller sends the asset and additional documents to the buyer. Escrow.com verifies that the buyer has received the asset correctly and the broker is notified as well. The buyer uses a predefined number of days to inspect the asset and either accept or reject it. In case the asset is accepted by the buyer, the seller and the broker are paid accordingly by escrow.com.

The flow diagram (src: escrow.com) below describes how the protocol works with Escrow.com.

A picture containing diagram

Description automatically generated

If the buyer rejects the asset, they are required to return it within 10 days. The Seller gets a 5-day period to inspect the returned asset. Based on the Seller's acceptance of the returned item, the Buyer is refunded the original payment less the entire escrow fee and any original shipping fees.

Involved assets include domain names, vehicles, electronic items, merchandise, jewelry, and watches.

Escrow.com provides multiple ways of broker transactions.

*Transparent to buyer:* transaction information such as asset price, seller’s proceeds and contact information can be visible to the buyer, seller, and broker.

*Transparent to seller:* financial terms (asset price, escrow fee, broker commission) are only visible to the seller.

*Transparent to buyer and seller*: financial terms are visible to all parties.

*Confidential:* financial terms are not visible to the buyer and seller. Both the broker commission and escrow fee can be paid by the Buyer, Seller, or split between the Buyer and Seller.

Managing Disputes:

In an event where the Buyer rejects the assets, they will use the seller's contact information to initiate a return. Escrow.com makes sure any complaints from either party are properly communicated, and that the Broker is kept fully informed at all times. In the event of a cancellation (after funds are received) or rejection of merchandise, the Buyer is responsible for the escrow fee regardless of whomever (Buyer, Seller, or Broker) is set up to pay the fee.

**2. EscrowGuardian:**

EscrowGuardian is a type of service that provides a secure, neutral third-party intermediary for financial transactions. It is typically used to facilitate the buying and selling of goods and services, particularly when one or both parties are located in different geographical locations. The escrow service holds onto the funds or assets being exchanged, and releases them to the appropriate party only after certain predetermined conditions have been met. This can help to reduce the risk of fraud and ensure that both parties are satisfied with the outcome of the transaction. EscrowGuardian is often used in real estate transactions, online marketplaces, and other situations where a neutral third-party is needed to hold and disburse funds.

EscrowGuardian holds onto the funds or assets being exchanged by placing them into a secure, segregated account. This account is typically managed by a neutral third-party, such as a bank or trust company, and is used specifically for the purpose of holding funds or assets in escrow. The account is often set up in such a way that the funds or assets cannot be released or transferred without the authorization of all parties involved in the transaction.

*B. Decentralized Applications*

**Smartlink:**

It is a blockchain-based application that executes commercial transactions through a decentralized escrow solution where smart contracts are implemented over the Tezos blockchain. Smartlink allows users to use a set of secure, audited, and pre-defined smart contract templates by utilizing the Smartlink smart contract library (described in the paper as Sscl). However, users can also customize or create new contracts specific to their requirements. The Smart contracts allow multiple stakeholders. Besides, they have a multi-step agreement protocol that allows involved parties to create several payment milestones. The involved parties in an escrow agree on a smart contract (from Sscl or create a new smart contract). The smart contract allocates an inspection period before the closure of a deal to ensure that the involved parties can inspect and confirm if the smart contract accommodates the prerequisites. After both parties agree with the terms defined in the smart contract the buyer provides some stipulated amount of shares (tez) that gets stored in the smart contract wallet. Next, the seller has to deliver the asset within a specific period of time. If both parties verify that all the conditions mentioned in the smart contract are satisfied, the escrow smart contract releases the withheld amount to the seller. A Mediator is potentially involved if there is a dispute between the buyer and the seller.

Smartlink provides the native token named SMAK. However, through the FA 1.2 standard, one can implement user-defined tokens as well. Smartlink also utilizes the FA2 standard to transfer NFTs.

*Smartlink Functionalities:*

A. Transferring Digital Asset:

The buyer and seller must connect their Tezos-compatible wallets to the Smartlink escrow services through the Smartlink API. There are 3 separate smart contracts involved: SMAK smart contract, Payment smart contract, and Escrow smart contract.

The whole procedure is as follows:

If the stakeholders agree and validate the transaction, and sign the transaction with their wallet, the Payment smart contract will create the payment and permit. Next, the Smartlink API will verify the payment data. The stakeholders will be notified regarding created payment in a separate email. If the created payment complies with the preset requirements then the involved parties will form a consensus.

The escrow can be made through the escrow smart contract if both stakeholders comply after they view and validate a transaction. To create an escrow, the parties will interact with the escrow smart contract. The permit is verified by connecting with the Payment smart contract and the escrow validity is confirmed by the Smartlink API. The stakeholders are then notified in separate emails and they become legally bound by the escrow.

B. Transferring Physical Asset:

The main difference between transferring physical assets and digital assets is that transferring physical assets potentially needs using and maintaining off-chain data. Furthermore, the involved parties may require end-to-end tracking of the physical asset.

The Smartlink Escrow Service utilizes several on-chain and off-chain mechanisms. The mechanisms verify the identity of the involved parties and allow end-to-end tracking of the physical assets. The verification of the identity of the involved parties is done by a built-in KYC that holds potential malicious actors responsible for any harmful behavior.

For the exchange of physical assets, there is always a shipping entity responsible for delivering the asset to the buyer. The shipping entity may or may not be the seller, and may rely on

subsidiary shipping services depending on circumstances.

The problem especially when multiple intermediate parties are involved is how to authenticate them, which may be dynamically changed during the shipping process.

Another challenge is collecting the accord from the final users for them to participate, and the proof that each intermediary complied with its corresponding function. It is to ensure the chain of custody of the goods being shipped, which is required in order to establish liabilities of the different intermediaries in case of disputes. The solution provided by Smartlink is to link up with identity providers and/or KYC providers to verify authentication, especially for the involved intermediate parties.

Interaction with SMAK, Payment, and Escrow smart contract is done on-chain and identity providing, KYC, and storing information on the shipment is done off-chain. Oracle is used to store off-chain states in the blockchain.

Smartlink allows both off-chain and on-chain models of authentication. In on-chain authentication dynamically providing identities from the off-chain world are stored on-chain.

Identity is either provided by a Certification Authority or by the Blockchain and Social Media Account Pair pattern.

In the first process, the subject requests a CA for the digital certificate. The certificate can be validated by any entity which is in possession of the public key of the CA. Once received, the subject forwards the certificate and the signed identity to the Adapter Smart Contract. An Oracle is used to check the validity of the certificate before posting it on-chain in the identifier registry.

In the second process, instead of involving a CA, the subject posts the generated public key along with the signed identity on a social media account and then forwards the link to the Adapter smart contract.

In the process of transferring custody in the exchange of physical assets, after the recipient is successfully authenticated by the Smartlink API, the Smartlink API generates a transaction to invoke the Escrow contract to query for the new custodian. Next, it is validated by the dealer's wallet and issued to the blockchain together with the public key of the recipient and with proof of the integrity of the asset.

In a scenario where the recipient does not accept the proof of integrity, or the asset is not delivered within the predefined time frame, the recipient will reject the custody of the asset and either of the parties can raise a dispute.

For dispute resolution, the system relies on a built-in chat feature dedicated to solving disputes on the platform and the potential intervention of a mediator if the parties can’t agree to the new terms. Besides, the Resolve Disputes Online (RDO) features an AI-enabled mediation module for high-volume dispute management.

*Important Features of Smartlink:*

a) Smartlink provides the native utility token SMAK over Tezos blockchain.

b) Smartlink provides on-chain decentralized governance.

c) Smartlink provides rewards to consumers for each transaction they have processed in SMAK.

*Limitations of the current escrow protocols:*

A. The truthfulness of the centralized protocol is based on the assumption that the centralized service provider is not an adversary.

B. For all the existing protocols the service is limited to buying and selling digital and physical assets. In our work, we will introduce a bigger domain and describe how the DSC can be useful for those use cases.

C. The existing applications allow only 1 moderator in case there is a dispute, which is pretty centralized even if the main application is decentralized (e.g. Smartlink). In DSC we will introduce the moderator set where multiple numbers of moderators will act to resolve a dispute.

D. In the existing applications, the moderator is only involved at the end of the operation. We believe, in order to allow moderators to properly resolve a dispute, moderators should be active in multiple stages. Hence we have introduced the idea of dividing the operation into atomic stages, where at the end of each stage there is a verifying stage by the moderators.

E. Based on the methodologies followed by existing applications, it is obvious that the buyer information, seller information, moderator information, and other relevant information such as information on the asset, and its delivery date are public (even if not public to any other entity, but public to all the involved parties). In DSC, all the user information is private, though they can verify their identity using zero-knowledge proof. Besides, due to dividing the operation into atomic stages, involved parties can include different moderator sets for different stages. Moderators involved in one specific stage can learn the sub-operation involved only in that stage, but not the whole operation.

*Further benefits that we want to provide with DSC:*

A. Similar to Smartlink, DSC also requires users to provide KYC during registration. Users need to log in to a centralized portal to access the application interface.

B. DSC will include the idea of Soul Bound Tokens (SBTs) which will be used to maintain a rating system for all the involved parties.

C. **Inclusion of Defi**: In DSC, other than the buyer entity, seller entity, and moderator set, a fourth type of entity set is the investor set. For collaborations where there is a transfer of assets (digital or physical), the buyer can request grants from the investor set. A separate smart contract will be responsible for creating the collaboration between the buyer and the investor that will include the total amount of the grant, the deadline to return, and the commission amount.

**Section 4: Decentralized Secure Collaborations (DSC)**

Decentralized Secure Collaboration or DSC, as the name suggests, is an application that provides secure collaborations among untrusted parties. While discussing probable collaborations between the buyer and seller we only talk about transferring assets (the features provided by current escrow applications). But we always overlook that there can be other types of use cases that may require untrusted parties to collaborate.

Some of such collaborations are as follows:

a) Freelancing Service: In this scenario, an entity may require a freelancing service from a second party. For example, the authority of a local coffee shop may ask a freelance web designer to create a website for them. Note that there is no transfer of assets, rather completing a certain objective. Though in both cases verification is needed (verification that the asset is transferred properly and the verification that the objective is completed properly).

b) Contract-Based Job: For scenarios where a company hires an employee for a certain job, it is needed to verify that the objective was done by the employee and the agreed payment was done by the company.

c) Influencer Marketing: Since social media has taken a giant leap in recent years, it is quite common among both big and small companies to collaborate with social media influencers to market their product. It is important to use a verifiable collaboration platform in this use case since scamming is becoming very common in this field, especially in the crypto world.

d) Crowdfunding: As we discussed earlier, a verifiable secure collaboration platform is needed for crowdfunding campaigns. If we consider dividing the campaign into atomic levels (each level being moderated by moderators) then not only it is possible to verify if the total target amount is reached, but parties can also agree on several other decisions (such as agreeing on increasing/ decreasing the target amount) based on parameters such as campaign initiator’s progress in the project during several stages of the campaign as well as the volatility of the market in different times throughout the campaign.

Definitions:

Since our application is more than import-export trading, we can't rely on the concepts of ‘buyer’ and ‘seller’. So in this subsection, we will redefine our entities and we will use these names for the rest of the paper.

a) Initiator: Instead of a buyer we have an initiator. An initiator is an entity that initiates a collaboration.

b) Acceptor: Instead of a seller we have an acceptor. An acceptor is an entity that accepts the collaboration request from an initiator.

c) Moderator Universe: It is the set of all the moderators available in DSC.

d) Moderator Set: It is the set that will monitor a particular collaboration. The moderator set is a subset of the moderator universe.

e) Investor Set: The investor set includes entities that can provide grants to the initiator for a collaboration.

f) Collaboration: It is the whole campaign between the initiator and the acceptor.

g) Atomicity: It is the property of dividing a collaboration into atomic levels.

In DSC, an initiator initiates a collaboration. If the initiator needs investments, then it needs to call - **Investor smart contract**. The smart contract will allow the investor to make a deal with one or more investors from the investor set. In case no investment is needed, the initiator can directly call the - **Collaboration smart contract**. When an acceptor accepts the collaboration smart contract, it will allow the initiator and acceptor to make a deal. Next, the initiator and the acceptor jointly can divide the collaboration into atomic levels and assign a moderator set for each atomic division. For this whole process, they need to call - **Moderator smart contract**. Note that there can be repetition of moderator sets for multiple atomic stages, and one single moderator set can be assigned for the whole collaboration. It is recommended to provide atomicity to the collaboration. But the initiator and the acceptor can also settle with a one-stage collaboration in which case only one moderator set will be responsible for verification, and similar to existing applications, the moderator set will verify the collaboration after it's completely done. If atomicity is provided to the collaboration then each atomic stage will be verified by its respective moderator set. Without this verification, the campaign cannot move to the next atomic stage. Further note that the involvement of moderators is spontaneous. It does not depend on the **‘dispute resolution call’** by either of the initiator or the acceptor.

If atomicity is provided and a different moderator set is assigned for different atomic stages, then a specific moderator set will only know about the details of the corresponding atomic stage, but not the whole collaboration. Besides, the initiator, acceptor, or moderators will have their identity kept secret from the other entities. In case one entity needs to verify its identity, it can provide a zero knowledge proof. We will discuss it later in detail.

In the next subsections, we will discuss in order the following topics before going to the overall pipeline.

a) Entity Registration and Entity Rating using SBT (SoulBound Tokens)

b) Investor Set

c) Moderator Universe, Moderator Set, and Atomicity

d) Dispute Resolution

*A. Entity Registration and Entity Rating using SBT:*

Before going further with entity registrations, we need to understand what Soul Bound Tokens are. The concept of SoulBound tokens (SBTs) was proposed by Vitalik Buterin, Glen Weyl, and Puja Ohlhaver, in May 2022 in the whitepaper, entitled “Decentralized Society: Finding Web3’s Soul”. The difference between SBT and NFT is that SBTs are non-transferable tokens representing a person’s identity. In other words, SBTs define the identity of a user, allowing individuals to verify all of their information. While NFTs can be traded among users (via different trading platforms such as Opensea and Rarible), SBTs can not be traded. Just like we cannot trade our Social Security Number or Passport with another person. The concept behind Soulbound Tokens seemingly comes from the game World of Warcraft (WoW). In the game, “soulbound” is a property of an item that prevents it from being traded or mailed to another character. SBTs can of course be updated and one user can hold multiple SBTs. For example, one user can have one type of SBT that represents her medical history while another type of SBT that represents her education history. As time passes, both kinds of SBTs may update (not necessarily though). SBTs can not be just ‘mined’ by users (which is possible for NFTs). Rather, SBTs are allocated by entities called Souls. For example, employee rating SBTs are provided by the corresponding company to its employees. SBTs can also be used in other applications such as representing work history, credit history, professional certifications and also user achievements. They can be tied to a myriad of other traits, features, and personal information such as verifying user identities like name, birthday, political affiliations, charitable giving, criminal record, nationality, religious upbringing, military history, and more.

Up until now, anyone can fill their resume (or brag on social media) with false information such as masters’ from an esteemed University. Currently, there is no proper way to verify the correctness of such information. But with SBTs, the esteemed University's “Soul” (which may be an administrator/controller of exams from the university) would have to grant the user's “Soul” an SBT of a diploma. In this respect, SBTs can be distributed amongst members of a group or institution as proof of affiliation. In this context, a user’s ‘soul’ actually signifies her private wallet. The validity of an SBT can be verified from the provider's soul. So it is not possible to claim false achievements and credentials. Since SBTs are put in the blockchain, it is also not possible to alter the information.

Now that we have the basic idea of SBTs covered, we can move forward with user registration, its rating system, and user identity verification with Zero Knowledge Proof in DSC.

The whole idea of entity registration is based on the idea of linking SBT ratings to user profiles in a way that user can provide their signatures along with any activity (initiating or accepting a collaboration, moderating a collaboration) without providing their private key. Anybody should be able to use the signature to retrieve the user’s public key. Since SBT ratings will be linked with the user’s public key, it will be possible to retrieve a user’s SBT rating if the signature is available. In case of disputes, any entity can request any other entity to provide proof of their identity. In that case, the requested entity must provide zero-knowledge proof - “I have a private key s1, a nonce s2 such that from them I can generate public key w1 and my user address w2, where I provide the w1, w2 as the public parameters”.

Note that the user registration platform will be centralized because the benefits provided by decentralization are not needed here. The user needs to provide certain information such as name, nationality, and identity proof in order to register in the system.

Upon registering, user will receive a public address (that will work as their identity in the DSC platform), a private key, and a public key. The private key is a random integer number with an upper bound of n (n is a large prime). The public key is generated from the private key. The user address is generated by hashing the private key appended with a nonce value provided by the user. A secure cryptographic hash function will be used to perform this operation. The difference between the user address and the public key, apart from the fact that the first one is an address and the second one is key, is that the user address will be used as the identity of the user in the DSC platform (similar to user’s wallet address; and will be used to call and interact with all the smart contracts) and the public key will be used to link up with the SBT ratings. However, one thing to note is that both the user address and the public key are created from the private key.

Public Parameters: User Address, Public Key

Private Parameters: Private Key, Nonce Value

Up next we will discuss in order:

1. Key and User Address Generation.

2. Signature Generation.

3. Verifying Signature and Linking with SBT Rating.

4. Identity Verification with Zero Knowledge Proof

Note that only the point 1 is a part of entity registration. Points 2 and 3 are parts of linking with SBT ratings. Point 4 is part of identity verification during disputes.

Diagram

Description automatically generated

Fig2: 1: User Registration, 2: Private Key is created. 3: Public Key is created from the private key. 4: User provides the Nonce Value. 5: Hash of Private Key and Nonce Value. 6: User Address is created

1. Key and User Address Generation:

We will use ECDSA for key generation.

The private key (privKey) is generated as a random integer in the range [0...n-1].

The public key (pubKey) is a point on the elliptic curve, calculated by the EC point multiplication:

pubKey = privKey \* G, where G is the generator point.

We will use the SHA512 algorithm to generate the user address. For this, the user needs to provide a secret nonce value also in the range [0...n-1].

The user address is generated by the formula:

User Address = SHA512[private key || nonce], where || defines the concatenation operation.

2. Signature Generation:

The algorithm takes a message value (**a unique nonce string generated by the initiator during the initiation of the collaboration**) along with the user’s private key and generates a signature. We will call the message value as **Collaboration Identifier**.

The algorithm works as follows:

a) Calculates the hash value of the message.

b) Generates securely a random number k in the range [1..n-1].

c) Calculate the random point R on the elliptic curve, where R = k \* G.

Its x-coordinate: r = R.x is noted.

d) Calculate the signature proof: *s* = k−1∗(h+r∗privKey)(mod n)

e) Return the signature {*r*, *s*}.

3. Verifying Signature and Linking with SBT Rating:

The algorithm takes the signed message and the signature {r, s} produced from the signing algorithm along with the public key pubKey, corresponding to the signer's private key as inputs. The algorithm returns the boolean True, iff the pubKey provided belongs to the same private key privKey that was used to generate the signature s. In any other case, the algorithm will return false.

For any new activity, users need to generate a new signature using their private key. Then they need to call a specific function called linker() which is available in all the smart contracts (except the SBT smart contract). The objective of the linker() function is to take the user address, the signature, the Collaboration Identifier, and the public key as arguments, check if the signature is unique, and then link them in a nested mapping as follows:

User address => {Collaboration Identifier => (public key, signature)}

We have discussed the linker() functionin the appendix section.

A getLinker()function will be used by any other entity to get the creator's signature and public key. For that, the entity needs to provide the signature creator’s user address and the Collaboration Identifier for the particular collaboration. The function will check if there is a corresponding entry, if so, will return the (public key, signature) pair.

The entity then can check the validity of the signature with the public key. If found valid, the entity can call the **SBT smart contract** with the received public key to get the SBT ratings of the signature creator.

Note that both signature creation and verifying of the signature will be done off-chain since they require complex calculations.

The signatures for a particular collaboration define the record of all the entities participating in the collaboration. The signatures will work as Non-repudiation proof since only the user with her valid (pubKey, privKey) pair can generate a valid signature.

4. Identity Verification with Zero Knowledge Proof:

In case of disputes, a user may need to prove her identity without providing her private key. In this case, the user will have to rely on providing a zero-knowledge proof.

The prover will have to prove that she holds the private key and nonce value such that the user address and the public key are generated from them.

It will work as follows:

The prover will generate a proof Q such that:

Q => User Address is SHA512 of privKey and the nonce value & pubKey is a point in the elliptic curve where pubKey = privKey \* G.

The prover will send the Q to the verifier.

The verifier can verify the validity of Q.

Note that, here we are relying on non-interactive zero-knowledge (NIZK) proof.

B. Investor Set:

In DSC, it is possible for initiators to initiate a collaboration without having enough funds. Since an acceptor needs to see enough locked funds before she can accept an initiation, the initiator needs to find investment prior to initiation. In other words, the **Investor smart contract** must be run before the **Acceptor smart contract**, in case there is a lack of funds.

The Initiator needs to summarize her project in her profile (required as both the acceptors and investors need to know the project details). Then the Initiator starts the Initiation Smart Contract. The Initiation Smart Contract can be used for notifying acceptors and investors. The constructor function of the smart contract will have a boolean argument: **investor**. This resembles if the Initiator wants investments and turning it on means applying for the funds. 2 separate global lists named initiation\_for\_acceptance\_list and initiation\_for\_funds\_list will be used to have the records of ongoing initiation deployed for either acceptance or investment.

This global list initiation\_for\_funds\_list can be accessed in the Investor Smart Contract by the investors. This will give them the opportunity of finding out which users have appealed for funds. User initiation will also be present off chain so that it can be shown off chain in the user profile in the front end.

Once the investors get the users’ ids of the users appealing for funds, they can visit their profiles, check their ratings and also read the description of the projects they are working on. The clarity in the project description by the users is necessary to convince the investors. Depending on all of these factors, now investors can decide if they want to invest in the project.

When the investor accepts the appeal, he and the initiator can make a deal about the investment policy. The deal will be done via another function in Investor Smart Contract. The function will include additional parameters the money being invested, the interest rate/other rates, and the timeline of payment. Upon accepting the deal by the initiator-investor duo, and starting the investor smart contract, a multi-signature wallet is created. For future reference, we will call it a multisig wallet.

Like this, the initiator can make deals with multiple investors to make a pool of investments. After that, she can reach out to the acceptor by calling the initiation constructor with the investor boolean set as false. Acceptor entities can know about the initiation from the initiation\_for\_acceptance\_list. Initiator’s total funds can be verified from her profile from the front end. Once the Acceptor Smart Contract is initiated and another multisig wallet is created. Now, the initiator, along with the investors, can authenticate so that from their multisig wallet, the fund reaches the multisig wallet between the acceptor and initiator.

Diagram

Description automatically generated

auth1, auth2, etc. here means authentications. For example, auth1 means that Initiator and Investor1 need to authenticate so that the fund from multisig wallet1 reaches the multisig wallet between initiator and acceptor. Once the fund leaves the multisig wallet1, its corresponding interest rate starts.

Now the question is why we need The Multisig Wallet among Initiator, Investor and Acceptor. This is so that all the parties can authenticate the flow of funds from the wallet to Acceptor’s personal wallet. This will eliminate any chance of fraud from Initiator+Acceptor against Investor. Every step there is a progress of the work by Acceptor, all of the 3 parties can authenticate it and agree to allow the flow of some funds from the Multisig wallet to Acceptor’s personal wallet.

Any time there is a dispute, Investor can stop further authentication and call for dispute resolution.

**Section 7: Appendix**

**Variables**:

mapping (address => mapping (string => (string, string))) public addressSBTLinker;

**Functions**:

**function linker (address userAddress, string signature, string collaborationIdentifier, string pubKey)**

{

Get global allSignatures[] array

If signature not in allSignatures[]

{

addressSBTLinker[userAddress][ collaborationIdentifier] = (pubKey, signature);

}

}

**function getLinker(address userAddress, string collaborationIdentifier) → (string, string)**

{

If addressSBTLinker[userAddress][ collaborationIdentifier] exists

{

return addressSBTLinker[userAddress][ collaborationIdentifier];

}

}

**constructor Initiation (bool investor)**

{

if (investor == true)

{

call investor\_initiation(useraddress);

}

else

{

call acceptor\_initiation(useraddress);

}

}

**function investing(address self\_address, address initiator\_address)**

{

var total\_amount;

var rate;

var timeline;

//setters for the variables

//deploy investment amount to multisig wallet

}

**REFERENCES:**

[1] <https://www.omnicoreagency.com/instagram-statistics/>

[2] <https://www.statista.com/statistics/1250681/share-of-instagram-influencers-involved-in-fraud-worldwide/>

[3]<https://transparency.fb.com/data/community-standards-enforcement/fake-accounts/facebook/>

[4] <https://about.linkedin.com/transparency/community-report>

[5] <https://www.techtarget.com/searchsecurity/feature/LinkedIn-scams-fake-Instagram-accounts-hit-businesses-execs>

[6] <https://www.phishlabs.com/blog/fraud-impersonation-fuel-q2-increase-in-social-media-attacks/>

[7] <https://www.ftc.gov/news-events/data-visualizations/data-spotlight/2022/01/social-media-gold-mine-scammers-2021>

[8] <https://www.cbsnews.com/news/influencer-marketing-fraud-costs-companies-1-3-billion/>

[9] <https://www.businesswire.com/news/home/20180515005775/en/Global-Brand-Counterfeiting-Report-2018-2020---ResearchAndMarkets.com>

[10] <https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/>

[11] <https://www.oberlo.com/statistics/how-many-people-shop-online>

[12] <https://www.paymentsdive.com/news/e-commerce-fraud-to-hit-20-billion-2021-an-18-jump-from-prior-year/599312/>

[13] <https://www.statista.com/topics/9240/e-commerce-fraud/#topicHeader__wrapper>

[14] <https://www.verifi.com/wp-content/uploads/2015/03/Verifi_wp_How-Much-Are-Chargebacks-Really-Costing-You-011515.pdf>

[15] <https://risk.lexisnexis.com/insights-resources/research/us-ca-true-cost-of-fraud-study>

[16] <https://www.cybersource.com/content/dam/documents/en/cybersource-ecommerce-fraud-explained-ebook-2020.pdf>

[17] <https://chargebacks911.com/chargeback-fraud/>

[18] <https://chargebacks911.com/chargeback-stats/>

[19] <https://www.forbes.com/sites/louiscolumbus/2020/05/18/how-e-commerces-explosive-growth-is-attracting-fraud/>

[20] <https://www.mastercardservices.com/en/recovery-insights/chronicles-new-normal-cybersecurity>

[21] <https://www.juniperresearch.com/press/ecommerce-losses-online-payment-fraud-48bn>

[22] <https://nilsonreport.com/upload/content_promo/NilsonReport_Issue1209.pdf>

[23] <https://www.crowe.com/global/news/fraud-costs-the-global-economy-over-us$5-trillion>