IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

REPUTABLE - A Decentralized Reputation System for Blockchain-based Ecosystems

JUNAID ARSHAD[1], MUHAMMAD AJMAL AZAD[2], ALOUSSEYNOU PRINCE[1], JAHID ALI[1] AND THANASIS G. PAPAIOANNOU[3]

[1]School of Computing and Digital Technology, Birmingham City University (email:junaid.arshad@bcu.ac.uk,[alousseynou.prince, jahid.ali] @mail.bcu.ac.uk)
[2]Department of Computer Science, University of Derby, UK (e-mail: m.azad@derby.ac.uk)
[3]Athens University of Economics & Business (AUEB) (e-mail:pathan@aueb.gr)

Corresponding author: Junaid Arshad (e-mail: junaid.arshad@bcu.ac.uk).

ABSTRACT Reputation systems are an important means to facilitate trustworthy interactions between on- and off-chain services and users. However, contemporary reputation systems are typically dependent on a trusted central authority to preserve privacy of raters or on adding noise into the user feedback. Moreover, the accuracy of reputation values relies on the integrity of user feedback or input; this feedback should not be tampered with or misused for other purposes. This paper presents blockchain-based reputation system named REPUTABLE (A Decentralized Reputation System for blockchain-based Ecosystems), which computes the reputation of service providers and external services within a blockchain ecosystem through decentralized on-chain and off-chain implementation. Specifically, REPUTABLE not only ensures privacy, but also reliability, integrity and accuracy of reputation values, while incurring minimal overhead. It also enables performing certain data or statistical analytics functions on user feedback, whilst preserving security, privacy, accountability and unlinkability of participants and their feedback. We present a proof-of-concept implementation and a demonstration of the REPUTABLE system. Finally, by means of formal and empirical evaluation, we show the effectiveness of our proposed system to preserve the anonymity of user feedback and the high performance of its blockchain-based implementation.

INDEX TERMS Reputation, Privacy, Unlinkability, Blockchain, Trust

I. INTRODUCTION

Blockchain is a disruptive paradigm that enables immutable transactions without the presence of a trusted third-party. Consequently, it has been adopted to achieve trustworthy applications across diverse domains such as healthcare, manufacturing, and finance. Blockchain-based applications require interaction with external (off-chain) services to avoid silos. However, as trustworthiness of transacting parties is uncertain, specific mechanisms to facilitate such transactions in a trustworthy manner are needed.

Trustworthiness is fundamental to the widespread use of services. Typically, the trustworthiness of a system or service associates a degree of reliability to the *Trustee* which serves as an indicator of its ability to perform specific functions. In this context, reputation systems can provide a measure of the trustworthiness of service providers. A typical reputation system utilises user feedback to evaluate the reputation of a service provider affecting the trustworthiness of the provider. Distributed Ledger Technologies (DLTs) such as blockchains are inherently decentralized peer to peer systems that bring together different nodes and sub-systems with typically no prior engagements. Therefore, trust becomes even more important within such environment to achieve reliable service provision. Blockchains deal with the challenge of trust through mechanisms such as consensus protocols and cryptographic foundations. However, as blockchain systems are being increasingly used to interface contemporary systems and services, the notion of trust can change relatively quickly, exposing a system to new threats which can adversely affect its trustworthiness.

Contemporary reputation systems involve a centralized authority to administer, aggregate and analyse user inputs to calculate the reputation of a service. Therefore, centralized approaches have inherent limitations including assigning too much trust and power to a central entity. Further, central authority within such systems can be biased, or can be compromised leading to potential data leakage. Furthermore, as reputation systems rely on user inputs, trustworthy management (collection, tracking, storage and processing) of such data feeds is critical to the overall effectiveness of a reputation system. Although blockchains are inherently decentralised a number of challenges exist in achieving reputation systems for blockchain-based ecosystems. For instance, although storing user feedback on blockchain ensures immutability, it can potentially compromise privacy of feedback whilst degrading scalability and at significant transaction costs. Further, authenticity, verifiability and transparency of reputation scores is critical to effectiveness of a reputation system however it is non-trivial and requires explicit efforts to achieve them.

This paper presents a decentralized, verifiable reputation system, REPUTABLE: A Decentralized Reputation System for Blockchain-based Ecosystems, which investigates the challenge of achieving trustworthy reputation calculation of services employing blockchain technology. Specifically, REPUTABLE is focused on achieving a trustworthy reputation system whilst preserving the security, privacy, accountability and unlinkability of participants and their responses. Further, we require that the decentralized reputation system makes conservative use of on-chain storage and smart-contract execution, so that it is scalable and cost-efficient. Moreover, the paper presents a proof of concept development of the REPUTABLE system in the ONTOCHAIN [1]blockchain ecosystem along with a formal and empirical evaluation with respect to anonymity of user feedback and performance of the blockchain implementation. Major contributions of this paper are:

- A blockchain-based decentralized reputation system for online marketplaces which takes into account user feedback to evaluate overall reputation of sellers whilst protecting user privacy and feedback anonymity. Through this, REPUTABLE adopts a user-centric approach which facilitates trustworthy service provision within blockchain-based ecosystems. Further, REPUTABLE leverages homomorphic cryptography to de-link user identities from feedback to achieve user privacy.
- REPUTABLE leverages cutting edge blockchain technologies to achieve its implementation which enables interoperability with emerging advancements within blockchain such as decentralised oracles, and side chains.
- We have conducted formal and empirical evaluation for REPUTABLE which highlights its ability to address specific requirements with respect to security privacy as well as performance efficiency.

The rest of the paper is organised as follows: Section II-A presents an overview of reputation systems highlighting fundamental practices and their role in achieving trustworthy systems. Section II-B analyses state of the art within this area and how REPUTABLE contributes to it. In Section III, we provide a detailed description of the REPUTABLE system along with its different components. The implementation details of the REPUTABLE system are presented in Section IV followed by a comprehensive evaluation of our approach in Section V. Finally, in Section VII, we conclude our work.

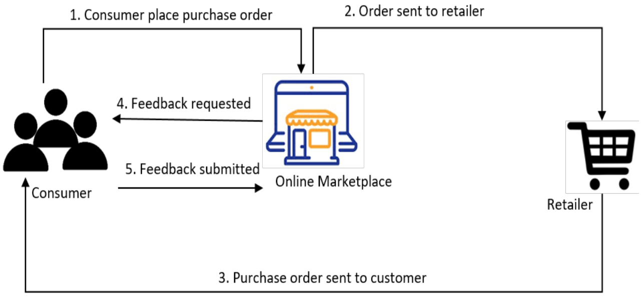

1. Consumer place purchase order
2. Order sent to retailer
4. Feedback requested
5. Feedback submitted
Consumer
Online Marketplace
Retailer
3. Purchase order sent to customer

FIGURE 1: Transaction and reputation workflow for a typical online marketplace

II. BACKGROUND AND RELATED WORK

In this section, we provide some basic background on reputation systems required to understand our proposed system and we overview the related work.

A. REPUTATION SYSTEMS

The marketplace (e-commerce, blockchain-based ecosystems, cloud marketplace etc.) facilitate their users to evaluate the trustworthiness service provider before consuming their resources. The function of these reputation systems requires feedback from users which are then aggregated together to compute the aggregate reputation of providers offering services [2], [3], [4], [5]. The reputation system can be either centralized [3], [6] or can operate in the decentralized settings [7], [8], [9], [10]. Figure 1 shows the flow of events that take place in the marketplace when a consumer submits the purchase order to a particular retailer. When the product is delivered to the consumer, the online marketplace asks the consumer for feedback against her recent interaction with the retailer. The user reports a feedback rating for the retailer and the marketplace then adds this trust score to the aggregated reputation of the retailer and displays this value on the web page designated for the retailer. The reputation systems require some feedback information from users, however, users might feel uncomfortable in providing such feedback because they concern about their privacy. Therefore the reputation systems are require to have inherent property of privacy, integrity and confidentiality of users involve in providing feedback scores or messages.

B. RELATED WORK

Several reputation systems have been proposed for different security settings. Hasan et al. [11] provided a comprehensive survey and evaluations of different privacy-preserving reputation systems. Pavlov et al.[12] proposed a secure multiparty aggregation model for aggregating the reputation scores from the participants in a malicious and honest but curious model. Kinateder and Pearson [13] propose a privacy-preserving framework that incorporates a trustworthy approach for shaping and accumulating sensitive feedback within the Trusted Platform of the node (TP). The participating nodes can demonstrate the trustworthiness and legitimacy of their identities without any confidential information about themselves. In this configuration, a trusted agent generates recommendations and determines what needs to be sent out anonymously to other nodes. Elan et al.[14] proposed a reputation scheme based on secure sum (utilizing a variety of techniques, including secret sharing). Li et al. [15] proposed a blockchain based data sharing and rewarding system for the Internet of Things. The system enables participants to participate in data sharing in an annoymous settings and earn the rewards without disclosing their real identity. Qi et al. [16] proposed a blockchain-based reputation systems that ensures the feedback anonymity and authenticity in order to compute the genuine reputation of users in the network. The system enables buyer to endorsed the feedback score of the seller thus identifies the fake scores. However, the system does not show resistance when buyer and seller collude with each other. Zhou et al. [17] proposed a blockchain-based decentralized reputation system for online marketplaces using interplanetary file system (IPFS)

Bag et al. proposed PrivRep [18], a customized privacy-aware decentralised reputation model for the electronic marketplaces. The system enables marketplace to compute the trust score of sellers, retailers or participants in a decentralized and anonymous way. The system utilizes a public bulletin board (PBB) which seems a centralized entity but can be implemented as the decentralized setup as suggested by the authors [19].

Several privacy-preserving systems have been proposed for crowd-sourcing setup in order to ensure the confidentiality, integrity and privacy of participants [20], [21], [22], [23]. Zhao [24] uses blockchain-based mobile crowdsensing to achieve privacy preserving reputation management. Differential privacy has also been employed in different scenarios but it introduce some noise that effect accuracy of results [25], [26], [27]. Several solutions have also been proposed for secure and privacy-aware aggregation of statistics [28], [29], [30], [31], [32]. A trust set of users have been employed as the relay agents for relaying the user scores [33] but the approach requires a trusted set of users which is difficult to employ in the real setup. Gibbs and Boneh [34] uses a small set of servers for performing the defined mathematical functions in a privacy-preserving way. The approach is based on the shared secret and requires small set of servers. Halevi et al. [35] proposed an aggregation scheme based on the homomorphic cryptosystem that evaluates the mathematical function securely and privately. However, the scheme requires PKI. Miao et al. [36] proposed a framework that performs a weighted aggregation over the user's encrypted data. The framework employs a homomorphic cryptosystem that has high accuracy in aggregation as well as protects the privacy of users.

With the increased use of blockchain technology, several efforts have been made to utilise the immutable property of blockchain ledger to aid recording provenance in a tamper-proof manner. ProvChain [37] represents one such effort where authors use blockchain to store cloud data provenance, i.e., metadata about cloud data objects. [38] presented a novel framework for evaluating the capability of innovative blockchain-based systems to deliver trustworthy recordkeeping based on archival science. The author presented a blockchain-based reference architecture to preserve the completeness, consistency, and naturalness of archival records. Although REPUTBALE shares the completeness and consistency characteristics with the application in focus in this paper, naturalness refers to events that are expected to occur as part of daily routine and not caused purposefully. The reference architecture presented is generic and does not address fine details with respect to data modelling and management. [39] presented a blockchain-based accountable method for data storage and processing. Focusing on big data applications, authors used a public blockchain-based auditing system that keeps a tamper-proof log of actions performed by participants. [40] is another effort to use blockchain's ability to provide tamper-proof storage to record data provenance. Specifically, the authors focus on the challenge of verifying the credibility of scientific experimentation results by recording and maintaining the provenance of such data. [41] proposed an audit mechanism that utilises oracles to record all transactions on the blockchain.

Blockchain technology has also been used to assure the privacy of users in an IoT network [42]. Chen et al. [43] designed a blockchain-based model to protect the privacy of participants in the big data environment. The system is more generally designed for protecting raw data but in our case, we protect the user's data while still performing some meaningful analytics over the encrypted data without actually decrypting it. Gan et al. [44] proposed a privacy-preservation model for task allocation in a crowd-sourced environment. Fortino et al. [45] designed a blockchain-based model to distribute the reputation score among nodes in a distributed IoT network. The proposed approach first computes the reputation of each node in the network and then develops the collaborative network among nodes for the network-wide view about the trustworthiness of nodes in the network as a whole. Tang et al.[46] proposed a protocol named IoT Passport that enables IoT devices from a different platform to collaborate with each other using the blockchain system. In this setup, the interaction between devices is signed with a digital signature and recorded in the temper-proof blockchain. A three-player game model is proposed in [47] that protects private information and friendship network of devices and users in the context of the connected social Internet of Things.

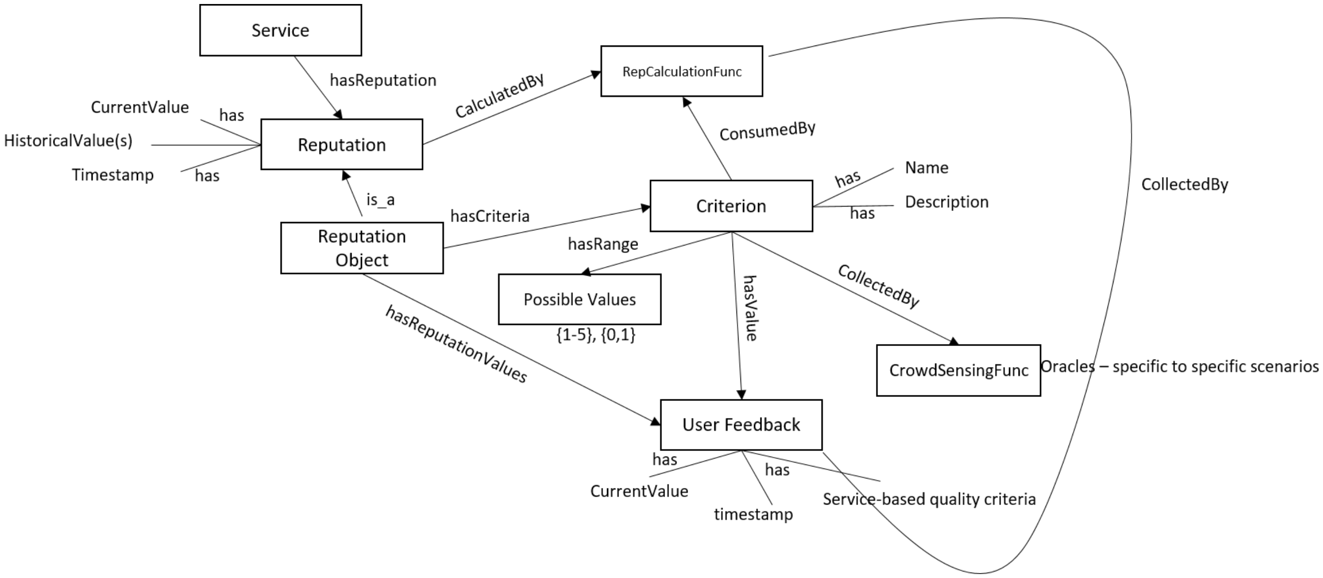


FIGURE 2: A high-level ontological structure for the reputation calculation mechanism

III. THE REPUTABLE SYSTEM

Our proposed solution takes a holistic approach to achieve a trustworthy, decentralized reputation model for blockchain-based ecosystems. A fundamental challenge is to ensure authenticity of feedback, i.e., feedback is not falsely inserted and only transacted parties can submit feedback. We address this challenge by use of tokens which are generated as a result of a qualifying transaction (purchase or service acquisition). Therefore each feedback is linked with a token to ascertain the validity and singularity of the feedback. Another major challenge is to gather and store individual user feedback that are used to calculate aggregate scores. Although all user feedback can be stored on the chain in the form of individual transactions, it has significant disadvantages such as the compromise on privacy, adverse impact on scalability, and cost incurred for storing feedback.

Furthermore, the reputation score should be transparent and verifiable to ascertain the correctness of the reputation calculation as well as to verify that all user feedback are included in calculating an aggregate reputation score. With respect to reputation modelling, we developing a verifiable reputation modelling mechanism that will enable evaluating the trustworthiness of external services whilst protecting user identities through homomorphic cryptography. Through the use of cryptographic primitives, an adversary would not be able to learn how a particular stakeholder has rated a particular user. Furthermore, we develop methods and interfaces to publicly verify reputation scores calculated by the reputation system.

The overall architecture of the proposed solution is presented in Figure 3 which provides an insight into the functioning of the proposed system whereas Figure 4 presents a typical transaction workflow for the proposed system. Details with regards to our design choices and different components of REPUTABLE system are presented below.

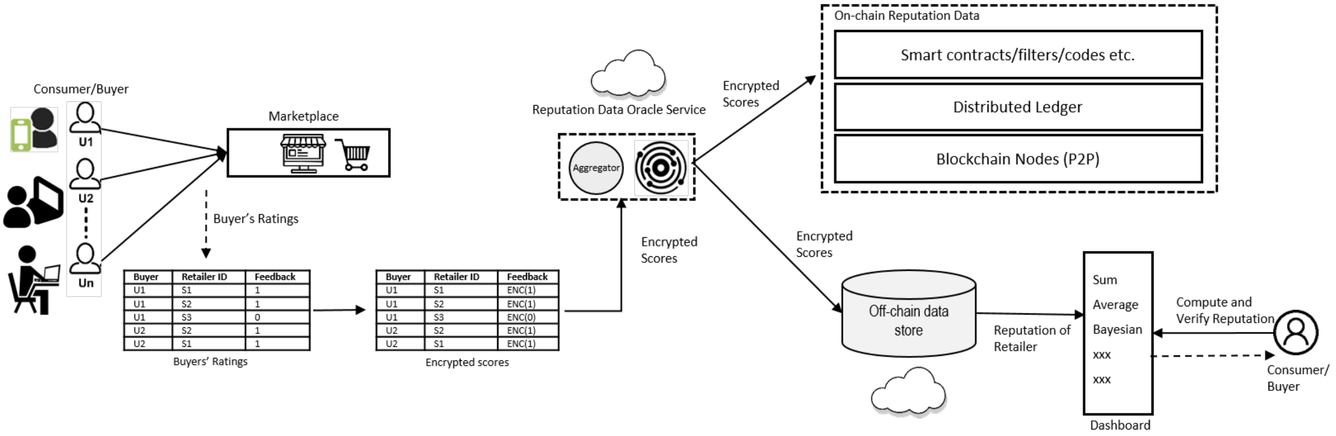
A. THREAT MODEL

In the privacy-preserving reputation system, our goal is to achieve following objectives: 1) to compute the trustworthiness of entity i.e seller or buyer while not disclosing their feedback scores because of fear of retaliation, and 2) computing the aggregate reputation by considering the scores that are within a prescribed range. Considering these objective, the threat model we lay down in our design involve two types of entities, the honest but curious entities- those who provide feedback in correct format but try to use the available information to breach privacy of others (learn value of feedback of targeted user), and second, malicious entities – who are there with the objective of manipulating and disrupting the functioning of the reputation system by providing out-of-range values. The proposed system defends against the threat model by encrypting the scores and utilizing non-interactive proofs to show the well-formedness of the encrypted scores.

B. REPUTATION MODELLING

The fundamental concept within REPUTABLE is that of reputation. In order to explain the concept of reputation as adopted within REPUTABLE, we present a high-level ontological structure for the reputation system in Fig. 2. The reputation is represented by an abstract entity Reputation Object which contains information about the reputation of a service. The reputation of a specific entity is therefore an instance of the Reputation Object and has a collection of attributes that together form the reputation score of service at a specific time instance. These include current values, historical values, and timestamps.

The *Reputation Object* comprises a number of criteria which are structured information points and can represent



Buyer	Retailer ID	Feedback
U1	S1	1
U1	S2	1
U1	S3	0
U2	S2	1
U2	S1	1

Buyers' Ratings

Buyer	Retailer ID	Feedback
U1	S1	ENC(1)
U1	S2	ENC(1)
U1	S3	ENC(0)
U2	S2	ENC(1)
U2	S1	ENC(1)

Encrypted scores

FIGURE 3: High-level architecture of the REPUTABLE system

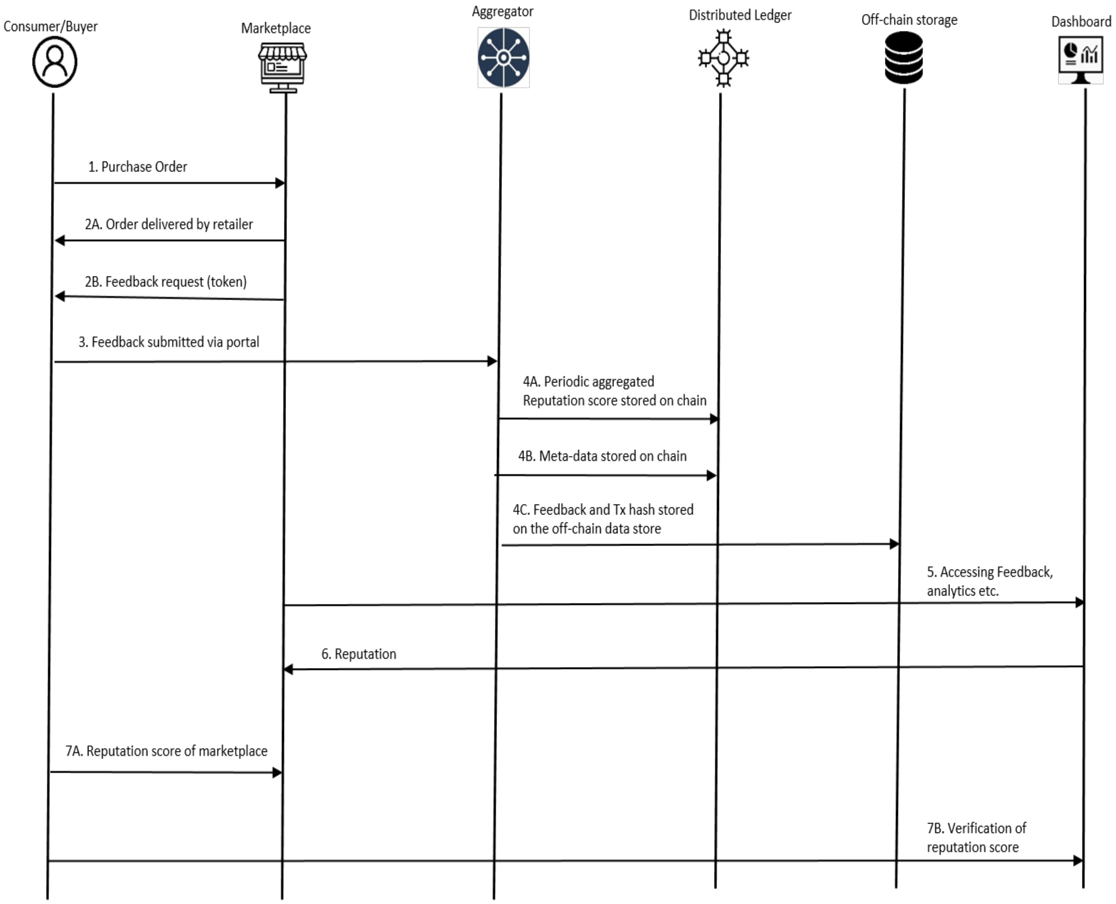

Consumer/Buyer
Marketplace
Aggregator
Distributed Ledger
Off-chain storage
Dashboard
1. Purchase Order
2A. Order delivered by retailer
2B. Feedback request (token)
3. Feedback submitted via portal
4A. Periodic aggregated Reputation score stored on chain
4B. Meta-data stored on chain
4C. Feedback and Tx hash stored on the off-chain data store
5. Accessing Feedback, analytics etc.
6. Reputation
7A. Reputation score of marketplace
7B. Verification of reputation score

FIGURE 4: A sample transaction workflow for the REPUTABLE system

elements of interest about a given service. Examples of criteria can be timeliness of service, driving style of a driver, quality/taste of apples etc. Each user assigns a specific value to a criterion based on their experience and is represented as user feedback. User feedback has a value (chosen by the user), timestamp and service-based quality criteria. Each criterion has a set of potential values which in the case of REPUTABLE are 0 or 1.

The user feedback is collected by a *CrowdSensingFunction* which is envisaged to be implemented in the form of a web form to facilitate usability. The *CrowdSensingFunction* relays the user feedback to the *ReputationCalculationFunction* which uses appropriate functions to calculate the reputation score for a service at a given time instance.

C. COMPONENTS OF REPUTABLE

In this section, we present different components of the REPUTABLE system along with their specific details.

1) Token Generation

One of the core concepts within any reputation system is the validity of the feedback or the question: How to determine whether a user is qualified to provide feedback on a service? Within REPUTABLE, we answer this question through the use of tokens. These tokens are issued by the seller against each valid purchase and are unique for a purchase. Once these tokens are generated, the seller distributes them to the qualifying customers through a communication medium such as email. As these tokens are generated by the seller, it makes this process susceptible to collusion attacks, i.e., collusion between the seller and the customer. We acknowledge this and consider this as an opportunity for future enhancements of REPUTABLE. For the current version of REPUTABLE, we assume seller and customer to be honest but curious.

2) User Engagement

Within the REPUTABLE system, a consumer of the marketplace is not required to anonymize his identity; instead, they hide their ratings by presenting cryptograms of ratings. The cryptograms, in this case, are encrypted feedback values where the encryption keys have been generated by the user. This feature achieves end-to-end decentralisation for the REPUTABLE system whilst also avoiding susceptibility to collusion by a central authority that may be responsible for cryptogram generation. The value of the rating score (0 or 1, like or dislike, rating between 1 to 5 stars) is encrypted using cryptographic primitives as shown in Figure 3. To this extent, the adversary on the reputation system or the reputation system itself would not be able to learn how a particular consumer has rated a particular retailer or another interacted consumer. The REPUTABLE system could provide maximum privacy unless a maximum number of consumers (n-1) in the system collude to find the rating score of the target consumer. Furthermore, the design choice of REPUTABLE ensures two other properties: 1) it limits consumers to provide rating scores within the prescribed range, and 2) it provides public verification of the reputation score stated by the marketplace.

3) Reputation Calculation

Once consumers have submitted their cryptograms and NIZK proofs to the bulletin board, any entity (participant, marketplace, or analyst) can compute the aggregated reputation of the retailer. Within REPUTABLE, this is performed by the aggregator component which has access to the feedback provided by the individual users. The process of reputation calculation is illustrated in the algorithm 1. We used reputation aggregation approch proposed by Azad et al. [19] with the objectives of implementing it over decentralized blockchain ecosystem. The system allows user to generate the public and private keys and publish the public key over the dashboard. The user who wants to contribute the feedback score computes the encryption keys using public keys of all participants from dashboard and encrypt the feedback score. The scores are published on the blockchain and aggregated in the privacy-preserving manner.

At this point, we already have the aggregate sum of positive ratings, i.e., the sum of consumers who have shown trust (1) in the retailer. The number of negative ratings can be computed by subtracting the positive ratings from the total number of users who have provided ratings. The simplest approach to compute the reputation of the retailer is to use the negative and positive ratings together, i.e., subtracting negative ratings from the positive ratings. We use the beta reputation system to compute the final aggregated reputation of the business entity or the retailer E on the marketplace. Let n be the number of consumers providing ratings, PE represents the number of consumers who provided positive ratings about entity E, and NE represents the number of consumers who rated the entity E as non-trustworthy, then the final reputation REE of an entity can be computed as follows: REE = (PE - NE)/ (n + 2)

The system can be easily extended to other reputation systems, e.g. the average of ratings can be computed by simply averaging the sum of individual ratings over the number of users. Cold-start problem limits new sellers to gain high reputation despite valid transactions this is because of small number of feedback scores. In our setup we address the problem by simply considering all users reputed for some specified number of transactions along with some fixed charges which is refundable after reputed behaviour over the number of transactions. This process would also limit fraudsters to use the system for malicious activities.The proposed system computes the aggregation in the centralized fashion so the convergence of the final reputation value will be 1 as all the feedback values requires for the computation of the reputation scores are all available in the centralized system [48]. However, one of the major challenges in such system is the selection of crowdsource users for participating in providing the feedback. In this system, we are using randomly selected set of trusted crowdsource users [49] who have atleast participated in previous aggregation process, however the system can be extended to consider the mix of trusted and non-trusted users which ultimately effects the final aggregated reputation and allow new users to participate in the aggregation process.

4) Dashboard

The dashboard is an important component of our proposed architecture. This component is envisaged to provide an interface to consumers (those providing feedback) and other interested third parties to query reputation scores. We envisage establishing this service off-chain potentially utilising cloud infrastructure both as a web-based endpoint as well as a programmable interface. The flexibility of having a programmable interface for the dashboard enables collaboration with other components of the architecture as well as external

Algorithm 1 Aggregating Reputation Scores

```
procedure AGGREGATE_REP_SCORES(ind_scores, campaign_id, seller_tokens)

    initialise $ind\_scores \leftarrow indiv\_user\_scores[user\_id]$
    initialise $seller\_tokens \leftarrow seller\_tokens[Array\_index]$
    initialise $campaign\_id$
    initialise $n\_responses$

    struct ind_score[ int user_score; string seller_address; int campaign_id; ]
    $ind\_scores[sellerHash] = ind_score$
    $store\_ind\_score(seller_i, ind\_score)$
    IF get_responseCount(seller$_i$, $campaign_j)equals$n$_responses$
        $retrieve\_ind\_scores(seller\_i, campaign\_j)$
     $aggregate(seller_i, campaign_j)$

   RETURN aggregate_score(seller$_i$)
```

services (such as reputation analytics) which can benefit from the output of the REPUTABLE system. Furthermore, an off-chain implementation also means that users will not have to install specific modules/plugins (MetaMask etc.) to access and interact with this service. Algorithm 2 presents an illustration of the proposed specification for the dashboard within REPUTABLE.

Algorithm 2 Dashboard

```
procedure DASHBOARD(user_id, campaign_id, seller_id)
    initialise struct user_scores (user_score, seller_addr, campaignID)
    initialise struct seller_aggr_score(tx_hash_aggr_score, timestamp, campaignID)
    initialise ind_scores:mapping(user_add=>ind_score)
    initialise tx_hash_aggr_score
    tx_hash = fetch tx_hash_aggr_scores(seller$_i$, $campaignID$)
  seller_aggr_score: calculate_aggr_rep_score(seller_id, ind_scores, aggr_func, campaignID)
  verify_aggr_rep_score(ind_scores, aggr_func, seller_id, campaignID)
  RETURN (Tx_hash, seller_aggr_score)
```

5) Blockchain and on-chain storage

Blockchain is a core component of the REPUTABLE system. It enables end-to-end decentralisation whilst also providing immutable, tamper-proof storage for reputation data (thereby facilitating trustworthiness and verifiability of reputation data). Within REPUTABLE system, reputation data consists of two different types. Firstly, it is the individual user feedback, i.e., the feedback provided by the users when contacted to share their experiences with a service/seller/marketplace. Secondly, it is the aggregate reputation score which is calculated using the individual user feedbacks. As these two types of data are linked with each other, we preserve this linkage and utilise it to achieve verifiable reputation scores. In addition to these two data types, REPUTABLE aims to capture important provenance information such as number of participants, number of responses, and timestamp etc. Such data is crucial to achieving the trustworthiness of the proposed reputation mechanism and verifiability of reputation scores.

The details of how reputation data is stored on-chain within REPUTABLE is illustrated by Algorithms 3 and Algorithm 4. Specifically, the proposed system stores the reputation data (aggregate reputation score) and its provenance in the form of transactions within the consensus blockchain through the execution of smart contracts.

Algorithm 3 Gateway Smart Contract

```
procedure GAETWAY_CONTRACT(ind_scores, campaign_id, seller_tokens)
    $initialisestructuser\_scores(user\_score, seller\_addr, campaignID)$
    $initialisestructuser\_token(user\_token, used, exp\_responses)$
    $initialiseseller_tokens : mapping(seller\_addrs => tokens)$
    $initialiseind\_scores : mapping(user\_add => ind\_score)$
     FOR seller_i in seller_list
         fetch aggregate$_reputation\_score(seller\_addr)$
      End FOR
      FOR user_score$_iinuser\_scores(seller\_addr == seller_j)$
          append (user_score,user_score_collated)
       End FOR
           store_ind_score_on_chain(seller_addr, user_score_collated, Campaign_k)

    RETURN Tx_hash
```

Algorithm 4 On-chain Storage Contract

```
procedure ONCHAIN_STORAGE(ind_scores, campaign_id, seller_id)
    $initialisestructuser\_scores(user\_score, seller\_addr, campaignID)$
    $initialisestructuser\_token(user\_token, used, exp\_responses)$
    $initialiseseller_tokens : mapping(seller\_addrs => tokens)$
    $initialiseind\_scores : mapping(user\_add => ind\_score)$
    Fetch_aggr_score(seller$_i$)
  Add_rep_score(seller$_i$, $ind\_scores$)

    RETURN Tx_hash
```

6) Off-chain storage and connectivity with blockchain

As highlighted earlier, we envisage storing raw user feedbacks on off-chain storage to facilitate user verification, querying, and interoperability. In addition to improving the scalability of the proposed solution, it also enables implementing a bespoke security layer to protect access to functions exposed by the REPUTABLE interfaces.

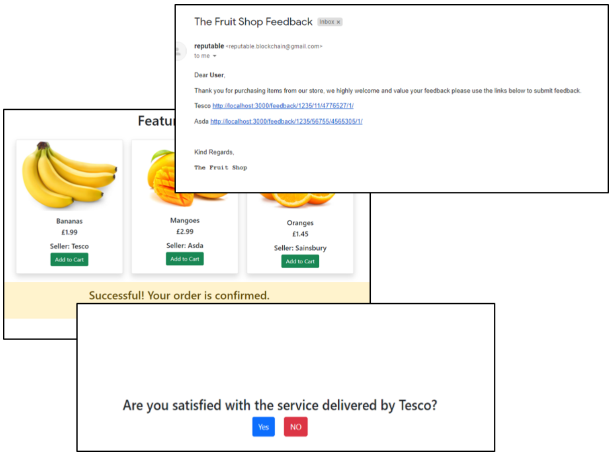
Furthermore, in order to achieve connectivity between on and off-chain components, we envisage using oracle for effective interoperability and linkage between on and off-chain storage. In this regard, the Reputation Data Oracle Service (RDOS) is envisaged to be responsible for managing the process of interacting with users to gather their feedback. RDOS achieves this by generating encrypted feedback data in accordance with the reputation model (0-5, 0,1, . . .). The details of the proposed design specification for this component is presented in algorithm 5.

Algorithm 5 Offchain Storage Contract

```
procedure OFFCHAIN_STORAGE(ind_scores, campaign_id, seller_id)
    $initialisestructuser\_scores(user\_score, seller\_id, campaignID)$
    $initialiseind\_scores : mapping(user\_add => ind\_score)$
    $initialisetx\_hash\_ind\_scores$
    tx_hash $\leftarrow fetchtx\_hash\_ind\_scores(seller_i, campaignID)$
      FOR $user\_score_i$ in user_scores(seller_id==seller$_i \&\& campaignID == campaignID_j$)
      store_ind_score_off_chain (seller_id, campaignID$_j$, $tx\_hash, user\_score)$
      End FOR
   RETURN Tx_hash
```

IV. IMPLEMENTATION AND SMART CONTRACTS

We have achieved a proof-of-concept implementation of the REPUTABLE system which leverages existing open source



The Fruit Shop Feedback

reputable <reputable.blockchain@gmail.com>
to me

Dear User,

Thank you for purchasing items from our store, we highly welcome and value your feedback please use the links below to submit feedback.

Tesco http://localhost:3000/feedback/1235/11/4776527/1/

Asda http://localhost:3000/feedback/1235/56755/4565305/1/

Kind Regards,

The Fruit Shop

Successful! Your order is confirmed.

Are you satisfied with the service delivered by Tesco?

FIGURE 5: User interfaces to facilitate feedback

technologies such as Ethereum Ropsten test network, Python, React, Firebase, python-paillier, and Remix. The PoC implementation of REPUTABLE comprises of: a user interface for feedback submission, smart contracts to process and store data on the chain, an aggregator to perform reputation aggregation, an oracle to facilitate interaction with the off-chain feedbacks, and a dashboard interface to facilitate querying and verification of feedback data and reputation scores. We present further implementation details of each component below.

A. USER INTERFACES

Our implementation provides two user interfaces, i.e., an interface to provide feedback, and a dashboard to query and verify individual and aggregate reputation scores as elaborated below.

- Feedback submission: This interface enables users to provide feedback for their respective purchases. Within the proof of concept implementation, we have used React to develop a web-based e-commerce marketplace to simulate user workflow presented in Fig. 4. Upon completing a purchase, a user is invited to submit their feedback answering a yesno question. Fig. 5 illustrates how a user can provide their feedback.
- Dashboard: The dashboard is implemented as a web-based interface that enables a user to achieve two tasks. Firstly, it allows a user to query the reputation score for a seller. This function utilises the individual scores for a seller that are stored in a Firebase off-chain data store and the API provided by the aggregator to calculate the aggregate reputation score for the selected seller. Secondly, the dashboard allows a user to verify the individual feedback provided to them for a seller. This function has been implemented using the API exposed by the aggregator and utilises user ID to identify individual users.

B. SMART CONTRACTS

The REPUTABLE system comprises of a number of smart contracts which are explained further below.

- *Data service smart contract*: In order to facilitate gathering user feedback from the web interface, we have implemented a data service in the form of a smart contract. This smart contract enables storing user feedback on the blockchain ledger to ensure the immutability of the feedback. Due to the module design of the REPUTABLE system, we believe this service presents an opportunity for further work by integrating sidechains to ensure scalability and performance efficiency. A detailed graphical illustration of this contract in the form of a flow chart is presented in Fig. 6.
- *On-chain data storage*: Within our implementation of the REPUTABLE system, we use a blockchain ledger to store individual user feedback, the aggregate reputation score for a seller, and meta-data for the reputation score. The individual user feedback is stored on-chain through the data service explained above however the storage of aggregate reputation score and provenance data is stored on-chain using the on-chain storage smart contract. Fig. 7 presents an code snippet for this smart contract with Fig. 8 presenting a graphical illustration of the function of the smart contract.
- *Off-chain data storage*: In order to facilitate the functionality provided by the dashboard, we have implemented a smart contract which stores individual user feedback using the Firebase cloud solution. This smart contract is triggered by the gateway to enable the storage of reputation data in a periodic manner.
- *Gateway contract*: The gateway smart contract provides a bridge between on-chain and off-chain data storage. Once the aggregate score for a seller has been calculated by the aggregator, the result is then stored on-chain via the gateway smart contract. The gateway contract contains a 'callback' function to which the oracle smart contract can interact with to send values which it itself has received from the off-chain oracle. This contract also can retrieve the aggregate score stored on-chain and cloud database URL respectively. A graphical illustration of the gateway smart contract is presented in Fig. 9

C. ORACLE SERVICE

The proprietary oracle consists of an oracle smart contract and an off-chain Python backend (illustrated in Fig. 10that serve the core functions of the oracle. The backend is responsible for performing the off-chain computation (aggregation) and sending the results back into the smart contract(s). This is done by utilizing the *Web3.py* library to listen for events emitted by the oracle smart contract and perform the necessary operation depending on which event was emitted. The oracle also establishes a connection to the cloud to store individual user feedback off-chain which are used by the dashboard

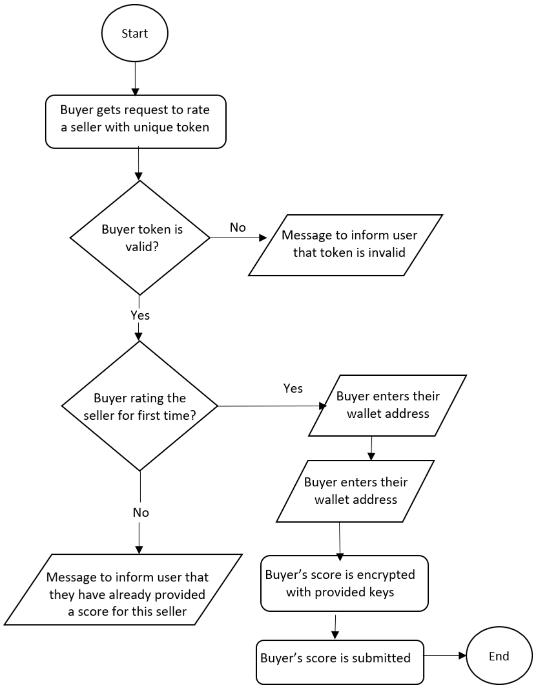
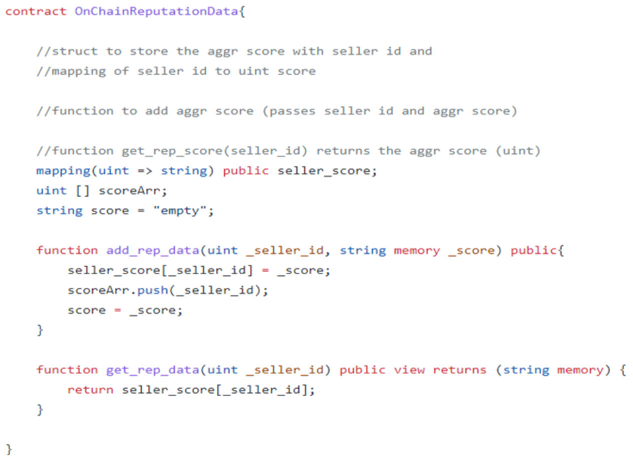

Start
Buyer gets request to rate a seller with unique token
Buyer token is valid?
No
Message to inform user that token is invalid
Yes
Buyer rating the seller for first time?
Yes
Buyer enters their wallet address
Buyer enters their wallet address
No
Message to inform user that they have already provided a score for this seller
Buyer's score is encrypted with provided keys
Buyer's score is submitted
End

FIGURE 6: Flow chart for data service smart contract



```
contract OnChainReputationData{

    //struct to store the aggr score with seller id and
    //mapping of seller id to uint score

    //function to add aggr score (passes seller id and aggr score)

    //function get_rep_score(seller_id) returns the aggr score (uint)
    mapping(uint => string) public seller_score;
    uint [] scoreArr;
    string score = "empty";

    function add_rep_data(uint _seller_id, string memory _score) public{
        seller_score[_seller_id] = _score;
        scoreArr.push(_seller_id);
        score = _score;
    }

    function get_rep_data(uint _seller_id) public view returns (string memory) {
        return seller_score[_seller_id];
    }

}
```

FIGURE 7: Smart contract for data storage on-chain

interface.

A part of the oracle smart contract is illustrated in Fig. 11. The oracle smart contract is responsible for sending a request and any required parameters to the off-chain python oracle. To accomplish this, this smart contract will emit an event depending on the service that the off-chain oracle is requested to do. There are two services that the off-chain oracle currently supports and that this smart contract will generate an event for:

- *Encryption of individual score*: This would emit the *RequestScoreEvent* which takes the parameters *sellerId* and *indi_score*. The results of each individual score would also be stored on Cloud Firestore
- *Aggregation of encrypted scores*: This would emit the

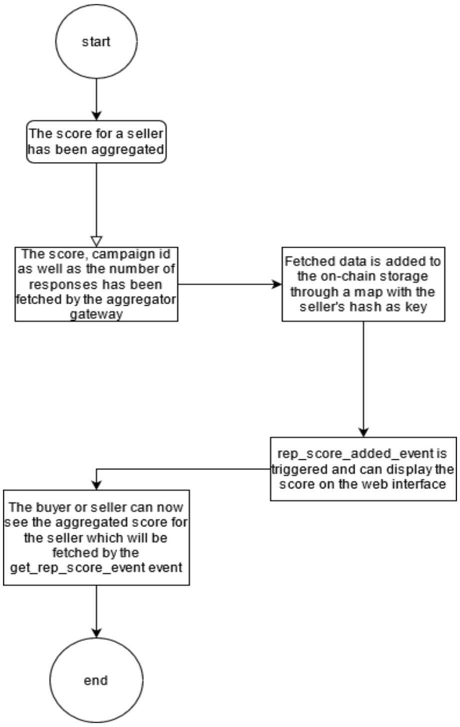

start
The score for a seller has been aggregated
The score, campaign id as well as the number of responses has been fetched by the aggregator gateway
Fetched data is added to the on-chain storage through a map with the seller's hash as key
rep_score_added_event is triggered and can display the score on the web interface
The buyer or seller can now see the aggregated score for the seller which will be fetched by the get_rep_score_event event
end

FIGURE 8: Flow chart for on-chain data storage smart contract

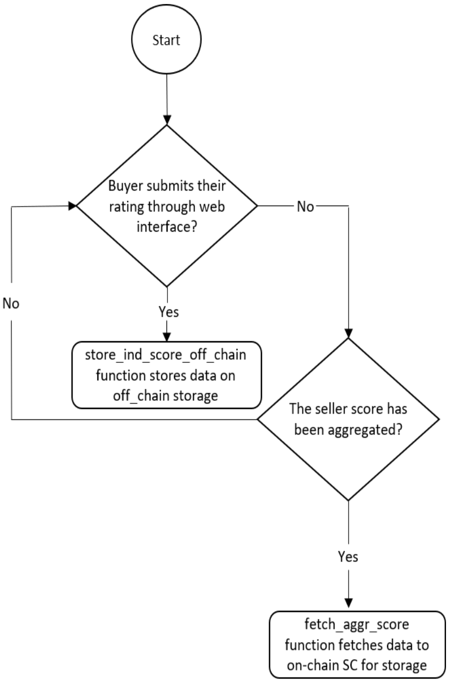
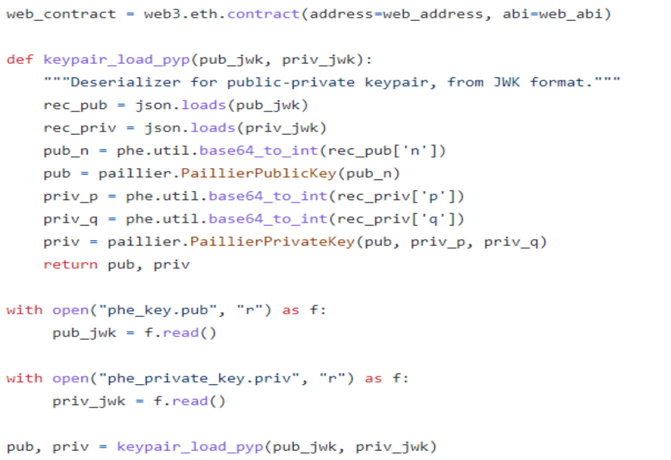
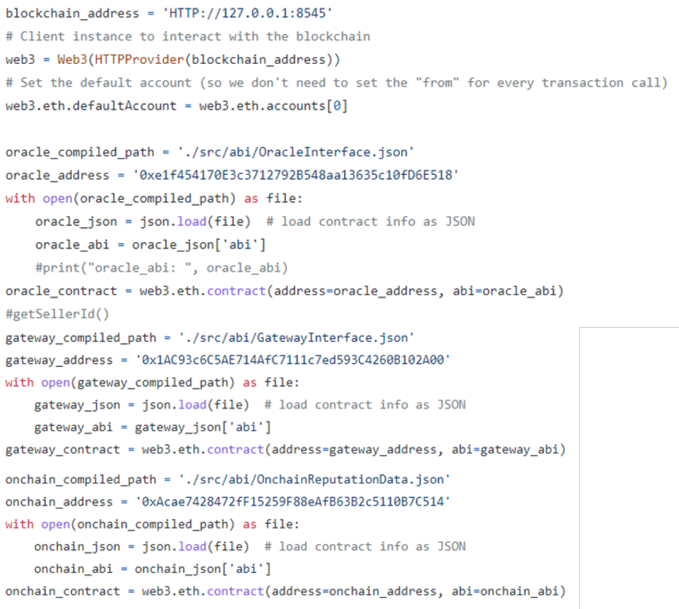

Start
Buyer submits their rating through web interface?
No
No
Yes
store_ind_score_off_chain function stores data on off_chain storage
The seller score has been aggregated?
Yes
fetch_aggr_score function fetches data to on-chain SC for storage

FIGURE 9: Flow chart for gateway smart contract



```
web_contract = web3.eth.contract(address=web_address, abi=web_abi)

def keypair_load_pyp(pub_jwk, priv_jwk):
    """Deserializer for public-private keypair, from JWK format."""
    rec_pub = json.loads(pub_jwk)
    rec_priv = json.loads(priv_jwk)
    pub_n = phe.util.base64_to_int(rec_pub['n'])
    pub = paillier.PaillierPublicKey(pub_n)
    priv_p = phe.util.base64_to_int(rec_priv['p'])
    priv_q = phe.util.base64_to_int(rec_priv['q'])
    priv = paillier.PaillierPrivateKey(pub, priv_p, priv_q)
    return pub, priv

with open("phe_key.pub", "r") as f:
     pub_jwk = f.read()

with open("phe_private_key.priv", "r") as f:
     priv_jwk = f.read()

pub, priv = keypair_load_pyp(pub_jwk, priv_jwk)
```

FIGURE 10: Feedback aggregation snippet



```
blockchain_address = 'HTTP://127.0.0.1:8545'
# Client instance to interact with the blockchain
web3 = Web3(HTTPProvider(blockchain_address))
# Set the default account (so we don't need to set the "from" for every transaction call)
web3.eth.defaultAccount = web3.eth.accounts[0]

oracle_compiled_path = './src/abi/OracleInterface.json'
oracle_address = '0xe1f454170E3c3712792B548aa13635c10fD6E518'
with open(oracle_compiled_path) as file:
    oracle_json = json.load(file)  # load contract info as JSON
    oracle_abi = oracle_json['abi']
    #print("oracle_abi: ", oracle_abi)
oracle_contract = web3.eth.contract(address=oracle_address, abi=oracle_abi)
#getSellerId()
gateway_compiled_path = './src/abi/GatewayInterface.json'
gateway_address = '0x1AC93c6C5AE714AfC7111c7ed593C4260B102A00'
with open(gateway_compiled_path) as file:
    gateway_json = json.load(file)  # load contract info as JSON
    gateway_abi = gateway_json['abi']
gateway_contract = web3.eth.contract(address=gateway_address, abi=gateway_abi)

onchain_compiled_path = './src/abi/OnchainReputationData.json'
onchain_address = '0xAcae7428472fF15259F88eAfB63B2c5110B7C514'
with open(onchain_compiled_path) as file:
    onchain_json = json.load(file)  # load contract info as JSON
    onchain_abi = onchain_json['abi']
onchain_contract = web3.eth.contract(address=onchain_address, abi=onchain_abi)
```

FIGURE 11: Snippet from oracle smart contract

RequestValueEvent which takes the parameters *campaignId* , *sellerId*, *userId* and *array*. The results of each aggregation would also be stored on Cloud Firestore

V. SECURITY ANALYSIS

A. VERIFIABILITY OF AGGREGATED SCORES AND PROVIDED FEEDBACK

In this proposed approach, the aggregated scores are stored in the off-chain and on-chain databases where the retailers, sellers and users can access this data to verify the computation scores, aggregated feedback values as well as individual feedback scores submitted by the users for products, sellers or retailers. The verifiability proves that the provided encrypted feedback scores well organized and are with the prescribe range. The system also provide mechanism of verifying the aggregate reputation score provided by the retailers to prove their trustworthiness and reputed behaviour over the platform.

B. PRIVACY AND INTEGRITY ANALYSIS

The system enables users to find out the aggregate scores of some retailers or sellers and they are able to do so using the feedback from the onchain and off chain databases. The major security measure is to ensure the privacy and integrity of feedback score provided by the users. The privacy of the users is ensured through the use of homomorphic encryption system where the feedback is formulated in such a way that this can only be revealed as an aggregate. The individual feedback would not provide any information about the likes and dislikes of individual feedback providers. The published feedback of user is in the range of 0 and 1 and it is formulated using the format $g^{xy}g^{v}$ for $v = 0$ or 1. The scheme is also secure if a number of feedback providers collaborate with each through the exchange of their keys to learn feedback of target users however if n-1 (n is the total number of users involve in submitting their feedback scores) user collude then they will learn score of remaining user. The retailers require to put the aggregate feedback score on their dashboard inorder to attract new users while showing their trustworthiness however nothing can be learned from this published aggregated score against some users.

The system ensures the security of feedback as the user feedback could only be used to find the aggregate reputation of seller or retailer and this would not reveal any sensitive information about the profile of the users. The system is dependent upon the trusted centralized system which is responsible for generation, secure exchange of keys and is not colluding with any other entity. In our computation we rely on the trusted setup and trusted channel which is being consider in many studies [50], [51].

VI. EMPIRICAL EVALUATION

We have conducted empirical analysis of our implementation of the REPUTABLE system. As part of this analysis, we have focused on assessing performance efficiency of individual functions, oracle, aggregator, and the dashboard. We have also varied number of users and number of feedback submitted by the users in our analysis.

A. EXPERIMENTATION SETUP

In order to conduct these experiments, our setup consisted of a web application, smart contracts, blockchain infrastructure, and off-chain (cloud) storage. The web application provided the user interfaces for feedback and dashboard functions and was a personal server hosted on a Microsoft Windows machine with 2.4GHz processor and 16GB RAM. The personal server also hosted the proprietary oracle which included the reputation aggregation function. The smart contracts were deployed on the Ethereum Ropsten testnet to enable use of Ethereum blockchain network for on-chain storage. Levering Ethereum Rospten testnet, we used the proof of work consensus algorithm available within the network. Further, Firebase

Benchmark	Value
size of the individual user feedback	128bytes
size of the aggregate score	128 bytes
Time taken to generate keys	62.5ms
Time taken to generate tokens	0.2ms
The time taken to store individual feedback to blockchain	15.62ms

Table 1: Benchmark performance parameters

was used as the off-chain cloud storage to store individual user feedback and achieve querying and verification through the dashboard interface.

B. DISCUSSION AND ANALYSIS

Table 1 presents the performance of specific functions such as the size of the individual user feedback and the time taken to store individual feedback to blockchain. For instance, an individual user feedback is represented using 128bytes which aids scalability of the approach, and also constitute that user is able to provide the feedback using ordinary mobile devices without any high data rate requirements.This also characterizes is even if user provide feedback for large number of users the bandwidth consumed is minimal. Further, the time taken by the data service provider or the system performing computation to store individual user feedback on blockchain is 15.62ms. This parameter is important as it can lead to delays in processing of individual feedback and calculation of aggregate reputation score. As shown in Table 1, this time is not significant and therefore facilitates the performance efficiency of the REPUTABLE system.

With respect to the performance efficiency of the oracle and the reputation aggregator, we have analysed the performance efficiency of these important components for varying number of users, i.e., 5, 50, 100, 500, and 1000. The outcome of the analysis for aggregator is presented in Figure 12 which demonstrates that although the time required for aggregation shows a steep rise when the number of users rise from 5 to 50 but then stabilise for the rest of the cases. This also shows the time is not substantial high for large number of data points. This demonstrates the ability of the REPUTABLE system to scale in an efficient manner even for large number of users providing feedback and number of feedback at the system. Similarly, as shown in Fig. 13, the time required by the oracle to store aggregated score on blockchain shows similar pattern, i.e., stabilising after initial rise which demonstrates the scalability of the REPUTABLE system even under huge number of users.

In addition to above, we analysed the efficiency of the dashboard component for querying and verifying reputation data (aggregate score and individual user feedback) for varying number of users, i.e., 5, 50, 100, 500, and 1000. As evident in Fig. 14, the time consumed by all three queries shows a steep increase when the number of users increase from 5 to 50 however this stabilises for the rest of the scenarios. This shows the scalability of the dashboard to support different queries across varying number of users.

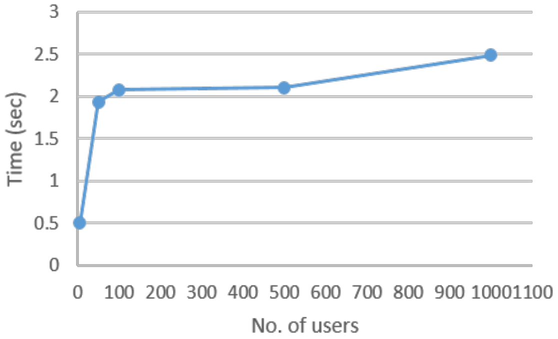


FIGURE 12: Time taken to calculate aggregate score

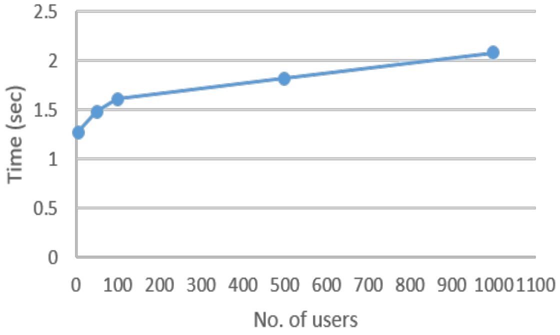
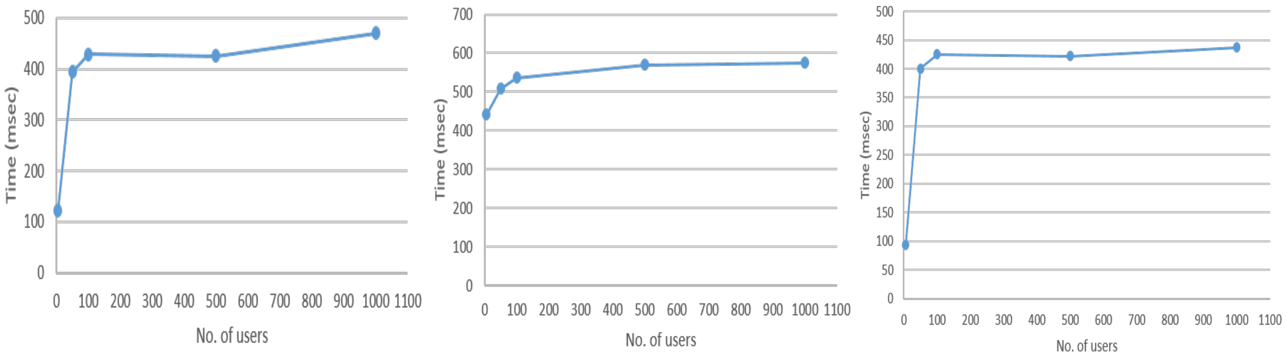


FIGURE 13: Time taken to store aggregate score on-chain

C. LIMITATIONS

From the experimentation and evaluation presented earlier, we have identified the following limitations to our solution which we envisage to explore as part of future work.

- The proprietary oracle is currently implemented as a centralised component which limits REPUTABLE to provide an end-to-end decentralised solution. We have done preliminary work with decentralised oracle services such as iEXEC [52] and Chainlink [53] however this work requires further effort to be incorporated within the REPUTABLE architecture.
- One of the critical components of the REPUTABLE system is the ability to store seller reputation scores on the blockchain. However, contemporary blockchain solutions introduce challenges such as scalability, transaction processing time, and financial cost of storing data on the chain. These challenges are envisaged to be aggravated for scenarios involving large number of users and feedback points. In this respect, a potential direction of future research is to explore using side-chains to achieve a scalable, efficient and cost-effective solution. Our efforts in this regard are at an initial stage and require further work.
- In the proposed system we randomly select the users to participate in the aggregation process which might effect the reputation aggregation process as it could



(a) time to query aggregate score for a seller

(b) time to verify aggregate score for a seller

(c) time to query individual feedback for a user

FIGURE 14: Dashboard performance (ms) for different queries across varying no. of users

include both legitimate and non-legitimate feedback provider which might increase or decrease the reputation of particular users. This approach also limits the new buyers to participate in the aggregation process. We are looking into to the approach which adopts the mechanism for fair selection of users from diverse groups in the selected crowdsource set.This can be bit challenging as feedback are completely encrypted and requires decryption for making reasonable decision.Another big challenge is identify users who are frequently changing the their identity to whitewash their previous reputation scores in order to rejoin the system as the new trusted users.

VII. CONCLUSION

We have presented our efforts to develop a decentralized, verifiable reputation system REPUTABLE which investigates the challenge of achieving trustworthy reputation of external services within a blockchain ecosystem. Specifically, REPUTABLE is focused at achieving trustworthy reputation system for external (off-chain) services whilst preserving security, privacy, accountability and unlinkability of participants and their responses. Along with a detailed description of the REPUTABLE system design and PoC implementation, we have presented formal and empirical evaluation with respect to anonymity of user feedback and performance of the blockchain implementation. The evaluation outcomes demonstrate the effectiveness and performance efficiency of the REPUTABLE system to achieve a decentralised reputation system.

We plan to continue our efforts to advance the REPUTABLE system by investigating the use of side chains as well as exploring time-window based reputation aggregation and assess the impact of these factors on the overall scalability of the REPUTABLE system.An additional area of future work is to explore the use of multiple feedback points. Specifically, current REPUTABLE implementation deals with one metric to represent user feedback however there are scenarios where multiple feedback points can be useful. For instance, a taxi service can be assessed based on punctuality, cost-effectiveness, and drivers' conduct. We envisage exploring such scenarios as part of future work.

VIII. ACKNOWLEDGEMENT

This research has received funding from the ONTOCHAIN project (ontochain.ngi.eu) which is funded by European Union's Horizon 2020 research and innovation programme under grant agreement No 957338.

REFERENCES

[1] "Ontochain: A new software ecosystem for trusted, traceable transparent ontological knowledge," 2022. [Online]. Available: https://ontochain.ngi.eu/

[2] L. De Alfaro, A. Kulshreshtha, I. Pye, and B. T. Adler, "Reputation systems for open collaboration," ACM Communication, vol. 54, no. 8, pp. 81–87, 2011.

[3] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," Elsevier Decision Support Systems, vol. 43, no. 2, pp. 618–644, 2007.

[4] F.Hendrikx, K.Bubendorfer, and R.Chard, "Reputation systems: A survey and taxonomy," Journal of Parallel and Distributed Computing, vol. 75, pp. 184 – 197, 2015.

[5] S. Bag, M. A. Azad, and F. Hao, "A privacy-aware decentralized and personalized reputation system," Computers & Security, vol. 77, pp. 514 – 530, 2018.

[6] L.Liu and M.Munro, "Systematic analysis of centralized online reputation systems," Elsevier Decision Support Systems, vol. 52, no. 2, pp. 438 – 449, 2012.

[7] M. Kinateder and S. Pearson, A Privacy-Enhanced Peer-to-Peer Reputation System. Springer Berlin Heidelberg, 2003, pp. 206–215.

[8] J. Bethencourt, E. Shi, and D. Song, "Signatures of reputation: Towards trust without identity," in Proceedings of the 14th International Conference on Financial Cryptography and Data Security, ser. FC'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 400–407.

[9] S. Schiffner, S. Clauß, and S. Steinbrecher, Privacy and Liveliness for Reputation Systems. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 209–224.

[10] M. A. Azad, S. Bag, and F. Hao, "M2m-rep: Reputation of machines in the internet of things," in Proceedings of the 12th International Conference on Availability, Reliability and Security, 2017, pp. 28:1–28:7.

[11] O. Hasan, L. Brunie, and E. Bertino, "Privacy-preserving reputation systems based on blockchain and other cryptographic building blocks: A survey," ACM Comput. Surv., vol. 55, no. 2, jan 2022. [Online]. Available: https://doi.org/10.1145/3490236

[12] E. Pavlov, J. S. Rosenschein, and Z. Topol, "Supporting privacy in decentralized additive reputation systems," in Trust Management, C. Jensen, S. Poslad, and T. Dimitrakos, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 108–119.

[13] M. Kinateder and S. Pearson, "A privacy-enhanced peer-to-peer reputation system," in E-Commerce and Web Technologies, K. Bauknecht, A. M. Tjoa, and G. Quirchmayr, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 206–215.

[14] E. Pavlov, J. S. Rosenschein, and Z. Topol, "Supporting privacy in decentralized additive reputation systems," in Trust Management, C. Jensen, S. Poslad, and T. Dimitrakos, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 108–119.

[15] T. Li, H. Wang, D. He, and J. Yu, "Blockchain-based privacy-preserving and rewarding private data sharing for iot," IEEE Internet of Things Journal, pp. 1–1, 2022.

[16] S. Qi, Y. Li, W. Wei, Q. Li, K. Qiao, and Y. Qi, "Truth: A blockchain-aided secure reputation system with genuine feedbacks," IEEE Transactions on Engineering Management, pp. 1–15, 2022.

[17] Z. Zhou, M. Wang, C.-N. Yang, Z. Fu, X. Sun, and Q. J. Wu, "Blockchain-based decentralized reputation system in e-commerce environment," Future Generation Computer Systems, vol. 124, pp. 155–167, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X21001850

[18] S. Bag, M. A. Azad, and F. Hao, "A privacy-aware decentralized and personalized reputation system," Computers Security, vol. 77, pp. 514–530, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404818304930

[19] M. A. Azad, S. Bag, and F. Hao, "Privbox: Verifiable decentralized reputation system for online marketplaces," Future Generation Computer Systems, vol. 89, pp. 44–57, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X17330315

[20] K. Yang, K. Zhang, J. Ren, and X. Shen, "Security and privacy in mobile crowdsourcing networks: challenges and opportunities," IEEE Communications Magazine, vol. 53, no. 8, pp. 75–81, August 2015.

[21] B. Rashidi, C. Fung, A. Nguyen, T. Vu, and E. Bertino, "Android user privacy preserving through crowdsourcing," IEEE Transactions on Information Forensics and Security, vol. 13, no. 3, pp. 773–787, March 2018.

[22] X. Jin and Y. Zhang, "Privacy-preserving crowdsourced spectrum sensing," in IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications, April 2016, pp. 1–9.

[23] S. Gao, X. Chen, J. Zhu, X. Dong, and J. Ma, "Trustworker: A trustworthy and privacy-preserving worker selection scheme for blockchain-based crowdsensing," IEEE Transactions on Services Computing, 2021.

[24] K. Zhao, S. Tang, B. Zhao, and Y. Wu, "Dynamic and privacy-preserving reputation management for blockchain-based mobile crowdsensing," IEEE Access, vol. 7, pp. 74 694–74 710, 2019.

[25] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in Theory of Cryptography, S. Halevi and T. Rabin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 265–284.

[26] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing," in Proceedings of the 23rd USENIX Conference on Security Symposium, ser. SEC'14. Berkeley, CA, USA: USENIX Association, 2014, pp. 17–32. [Online]. Available: http://dl.acm.org/citation.cfm?id=2671225.2671227

[27] Z. Zhang, S. He, J. Chen, and J. Zhang, "Reap: An efficient incentive mechanism for reconciling aggregation accuracy and individual privacy in crowdsensing," IEEE Transactions on Information Forensics and Security, vol. 13, no. 12, pp. 2995–3007, Dec 2018.

[28] U. Erlingsson, V. Pihur, and A. Korolova, "Rappor: Randomized aggregatable privacy-preserving ordinal response," in Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, ser. CCS '14. New York, NY, USA: ACM, 2014, pp. 1054–1067. [Online]. Available: http://doi.acm.org/10.1145/2660267.2660348

[29] H. Polat and W. Du, "Privacy-preserving collaborative filtering using randomized perturbation techniques," in Third IEEE International Conference on Data Mining, Nov 2003, pp. 625–628.

[30] A. Broder and M. Mitzenmacher, "Network applications of bloom filters: A survey," Internet Mathematics, vol. 1, no. 4, pp. 485–509, 2004. [Online]. Available: https://doi.org/10.1080/15427951.2004.10129096

[31] Z. Erkin, T. Veugen, T. Toft, and R. L. Lagendijk, "Generating private recommendations efficiently using homomorphic encryption and data packing," IEEE Transactions on Information Forensics and Security, vol. 7, no. 3, pp. 1053–1066, June 2012.

[32] M. Azad, S. Bag, S. Tabassum, and F. Hao, "privy: Privacy preserving collaboration across multiple service providers to combat telecoms spam," IEEE Transactions on Emerging Topics in Computing, pp. 1–1, 2017.

[33] Z. Wang, X. Pang, Y. Chen, H. Shao, Q. Wang, L. Wu, H. Chen, and H. Qi, "Privacy-preserving crowd-sourced statistical data publishing with an untrusted server," IEEE Transactions on Mobile Computing, pp. 1–1, 2018.

[34] H. Corrigan-Gibbs and D. Boneh, "Prio: Private, robust, and scalable computation of aggregate statistics," in Proceedings of the 14th USENIX Conference on Networked Systems Design and Implementation, ser. NSDI'17. Berkeley, CA, USA: USENIX Association, 2017, pp. 259–282. [Online]. Available: http://dl.acm.org/citation.cfm?id=3154630.3154652

[35] S. Halevi, Y. Lindell, and B. Pinkas, "Secure computation on the web: Computing without simultaneous interaction," in Advances in Cryptology – CRYPTO 2011, P. Rogaway, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 132–150.

[36] C. Miao, W. Jiang, L. Su, Y. Li, S. Guo, Z. Qin, H. Xiao, J. Gao, and K. Ren, "Privacy-preserving truth discovery in crowd sensing systems," ACM Trans. Sen. Netw., vol. 15, no. 1, pp. 9:1–9:32, Jan. 2019. [Online]. Available: http://doi.acm.org/10.1145/3277505

[37] X. Liang, S. Shetty, D. Tosh, C. Kamhoua, K. Kwiat, and L. Njilla, "Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability," in 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID). IEEE, 2017, pp. 468–477.

[38] V. L. Lemieux, "Blockchain and distributed ledgers as trusted recordkeeping systems," in Future Technologies Conference (FTC), vol. 2017, 2017.

[39] C. S. Nasikas, "Accountable and privacy preserving data processing via distributed ledgers," 2018.

[40] A. Ramachandran, D. Kantarcioglu et al., "Using blockchain and smart contracts for secure data provenance management," arXiv preprint arXiv:1709.10000, 2017.

[41] J. C. López-Pimentel, O. Rojas, and R. Monroy, "Blockchain and off-chain: A solution for audit issues in supply chain systems," in 2020 IEEE International Conference on Blockchain (Blockchain). IEEE, 2020, pp. 126–133.

[42] S. Moin, A. Karim, Z. Safdar, K. Safdar, E. Ahmed, and M. Imran, "Securing iots in distributed blockchain: Analysis, requirements and open issues," Future Generation Computer Systems, vol. 100, pp. 325 – 343, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X18330851

[43] Y. Chen, H. Xie, K. Lv, S. Wei, and C. Hu, "Deplest: A blockchain-based privacy-preserving distributed database toward user behaviors in social networks," Information Sciences, vol. 501, pp. 100 – 117, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025519305250

[44] X. Gan, Y. Li, Y. Huang, L. Fu, and X. Wang, "When crowdsourcing meets social iot: An efficient privacy-preserving incentive mechanism," IEEE Internet of Things Journal, pp. 1–1, 2019.

[45] G. Fortino, F. Messina, D. Rosaci, and G. M. L. Sarne, "Using blockchain in a reputation-based model for grouping agents in the internet of things," IEEE Transactions on Engineering Management, pp. 1–13, 2019.

[46] B. Tang, H. Kang, J. Fan, Q. Li, and R. Sandhu, "Iot passport: A blockchain-based trust framework for collaborative internet-of-things," in Proceedings of the 24th ACM Symposium on Access Control Models and Technologies, ser. SACMAT '19. New York, NY, USA: ACM, 2019, pp. 83–92. [Online]. Available: http://doi.acm.org/10.1145/3322431.3326327

[47] K. Li, L. Tian, W. Li, G. Luo, and Z. Cai, "Incorporating social interaction into three-party game towards privacy protection in iot," Computer Networks, vol. 150, pp. 90 – 101, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128618312945

[48] K. Walsh and E. Sirer, "Experience with an object reputation system for peer-to-peer filesharing," pp. 1–1, 01 2006.

[49] M. Allahbakhsh, A. Ignjatovic, B. Benatallah, S.-M.-R. Beheshti, E. Bertino, and N. Foo, "Reputation management in crowdsourcing systems," in 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2012, pp. 664–671.

[50] L. Melis, G. Danezis, and E. D. Cristofaro, "Efficient private statistics with succinct sketches," In NDSS, 2016.

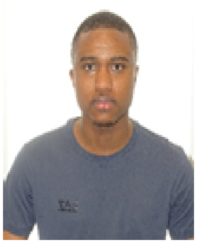
[51] M. A. Azad, S. Bag, S. Parkinson, and F. Hao, "Trustvote: Privacy-preserving node ranking in vehicular networks," IEEE Internet of Things Journal, vol. 6, no. 4, pp. 5878–5891, 2019.

[52] "iexec blockchain-based decentralized cloud computing," 2022. [Online]. Available: https://iex.ec

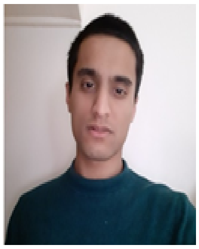
[53] "Chainlink: Blockchain oracles for hybrid smart contracts," 2022. [Online]. Available: https://chain.link/

JUNAID ARSHAD is an Associate Professor in cybersecurity at the Birmingham City University, UK. Junaid achieved his PhD from the University of Leeds, UK where he investigated the challenge of effective intrusion severity analysis for clouds. His research is focused at challenges within cyber security emphasising impact of novel and emerging technological paradigms, such as blockchain, distributed systems, cloud computing and big data. He has been actively involved in publishing high quality research within this field and has served on Program and Review Committee of a number of journals and conferences.

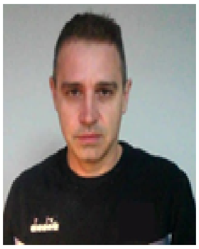
MUHAMMAD AJMAL AZAD is a Senior Lecturer (Assistant Professor) in Cyber Security in the Department of Computer Science and Mathematics at University of Derby. His research interests broadly cover the areas of network security and privacy. In the past few years, he designed systems and methods for securing the telecommunication users from the telemarketers, robo-callers, scammers, and spammers using behavioural modelling and social network analysis. He also worked in privacy and security of blockchain ecosystems and securing users over the online social networks. Muhammad has more than 80 publication in reputed journals and conferences. Muhammad served as an Associate Editor of Wiley Communication security for more than 3 years and also served a PC members for many reputed conferences for example Golbecomm, ICC, PETS etc.



ALOUSSEYNOU PRINCE has achieved an MSc in Big Data Analytics from Birmingham City University. His individual Master's project focused on Natural Language Processing to automate project classification and assignment based on year of study and other properties. He has been focusing lately on projects relating to software engineering or Artificial Intelligence.



JAHID ALI is currently an undergraduate student at Birmingham City University studying Cyber Security BSc. His primary research interests include working with contemporary and emerging blockchain solutions and their applications in solving real world problems. His current research includes working with blockchain technology to improve traceability within the supply chain.



THANASIS PAPAIOANNOU received his B.Sc. (1998) and M.Sc. (2000) in Computer Science from the University of Crete, Greece, and his Ph.D. (2006) in Informatics from Athens University of Economics and Business (AUEB). Since January 2006, has been a Senior Researcher with STEcon lab at AUEB. In parallel, since September 2016, he has been teaching in the University of Thessaly, Python Programming and Android Programming. Since December 2016, he has been also giving lectures in the Technological Educational Institute of Thessaly on Data Management in Internet of Things. Formerly, from 2013 to 2016, he has been with the Networking Lab of the Center for Research and Technology - Hellas (CERTH), Greece. Prior to that, from 2008 to 2013, he has been a postdoctoral researcher at the Distributed Information Systems laboratory (LSIR) of the Ecole Polytechnique Federale de Lausanne (EPFL). He has worked as project manager and scientific contributor in 17 national and European projects. He has authored more than 65 research papers in high-quality journals and conferences, which have received particular attention by the scientific community with more than 1200 citations.