EthReview: An Ethereum-based Product Review system for Mitigating Rating Frauds

Maryam Zulfiqar, Filza Tariq, Muhammad Umar Janjua,
Adnan Noor Mian, Adnan Qayyum, Junaid Qadir, Falak Sher,
Muhammad Hassan

PII:	S0167-4048(20)30367-9
DOI:	https://doi.org/10.1016/j.cose.2020.102094
Reference:	COSE 102094

To appear in: *Computers & Security*

Received date:	27 February 2020
Revised date:	19 September 2020
Accepted date:	21 October 2020

Please cite this article as: Maryam Zulfiqar, Filza Tariq, Muhammad Umar Janjua, Adnan Noor Mian, Adnan Qayyum, Junaid Qadir, Falak Sher, Muhammad Hassan, EthReview: An Ethereum-based Product Review system for Mitigating Rating Frauds, *Computers & Security* (2020), doi: https://doi.org/10.1016/j.cose.2020.102094

EthReview: An Ethereum-based Product Review system for Mitigating Rating Frauds

Maryam Zulfiqar[1,*], Filza Tariq[1], Muhammad Umar Janjua, Adnan Noor Mian, Adnan Qayyum, Junaid Qadir, Falak Sher, Muhammad Hassan

Information Technology University of Punjab (ITU), Lahore, Pakistan

Abstract

On an e-commerce platform, a buyer‘s decision about the purchase of products is highly dependent on the existing reviews of that product. These reviews provide a judgment metric about the quality of the products and the credibility of the sellers. Popular e-commerce platforms (like eBay, Amazon, etc.) receive and store a lot of customer reviews regularly. However, these e-commerce platforms are prone to various kinds of rating frauds (intended towards bad-mouthing and ballot-stuffing) carried by the fake buyers and fraudulent sellers to promote or demote certain products. Moreover, these e-commerce platforms are under the control of a single authority and are prone to tampering of the reviews by the centralized authority itself to increase product sales. These fake and tampered reviews lead towards bad online buying experience and a lack of trust in e-commerce platforms. In this paper, we propose a decentralized Ethereum based product review system, *EthReview*, which maintains the integrity of product reviews and is resilient against rating frauds. Our solution proposes a consortium network of randomized peer-to-peer (P2P) endorser nodes for the validation and verification of the truthfulness of product reviews. The honest reviewers are given incentives in the form of discount tokens for encouraging honest behavior in the system while the fraudulent behavior is penalized. We also provide security and performance analysis of EthReview to evaluate its robustness and resilience against different forms of rating fraud.

Keywords: Blockchain, Product Review Systems, Ethereum, Rating Fraud, dApp

1. Introduction

Electronic commerce (e-commerce) refers to the conduct of commercial activities including the sales and purchase of products and services over the Internet. In contemporary e-commerce systems, feedback systems are deployed to model the reputation and trust associated with sellers and their listed products. Such review-based reputation systems are extensively used on platforms like eBay[2], Amazon[3], Airbnb[4], etc.

It is documented in the literature that the feedback left by previous buyers via online reviews significantly impacts the choice of products of future buyers and on the sales growth of products [1]. In June 2017, as per a survey on behaviors of US online consumers [2], 63% of the respondents agreed that it was extremely important to read reviews about the product before its purchase. Another survey, conducted in March 2017 [3], illustrates the impact of user-generated online reviews on the purchasing decision of the online buyers.

Keeping the importance of reviews in mind, we believe that one of the fundamental design features of an effective product review system is the maintenance of the integrity of reviews and prevention against the injection of fake reviews. Fake reviews can lead to many problems in e-commerce systems including the defrauding of customers, encouragement of dishonest behavior, harming of healthy competition, the setting of perverse incentives, and the erosion of trust in e-commerce.

We believe that the fundamental problem with existing product review systems is their centralized underlying architecture [4, 5, 6, 7, 8, 9, 10], which makes such systems vulnerable (as the whole system can be brought down by compromising the single point of control). In such centralized architecture, a central authority or a trusted third-party is always required for review validation, verification, and managing refund claims [11, 12]. However, the involved central authorities can potentially filter, tamper, add, or reject the reviews based on their preference

[*] Corresponding author

Email addresses: maryam.zulfiqar1@itu.edu.pk (Maryam Zulfiqar), filza.tariq@itu.edu.pk (Filza Tariq), umar.janjua@itu.edu.pk (Muhammad Umar Janjua), adnan.noor@itu.edu.pk (Adnan Noor Mian), adnan.qayyum@itu.edu.pk (Adnan Qayyum), junaid.qadir@itu.edu.pk (Junaid Qadir), falak.sher@itu.edu.pk (Falak Sher), hassan.raza@itu.edu.pk (Muhammad Hassan)

[1] Both authors have equal contribution

[2] https://www.ebay.com/

[3] https://www.amazon.com/

[4] https://www.airbnb.com/

[13, 14, 15, 16], thus putting a question mark on the credibility of the product reviews. Such systems are also prone to rating frauds by fraudulent sellers and buyers, for e.g., ballot-stuffing [5], bad-mouthing [6] [8, 17, 18, 19, 20, 21, 15]. Further, they may suffer from problems such as fake refund claims and collusion of sellers and fake buyers to inject fake reviews to promote or demote products.

All of these vulnerabilities give rise to concerns over the validity and integrity of the posted reviews. Henceforth, there is a need to find an alternative to traditional product review systems for ensuring review integrity and truthfulness and also to combat rating frauds intended to perform feedback abuse, opinion spam, or fake claims.

In recent years, blockchain has emerged as a major technology for building tamper-proof decentralized systems [22, 13]. In this paper, we propose an Ethereum blockchain-based peer-to-peer (P2P) product review system known as the *EthReview*. The system is resilient against rating frauds prevalent in existing traditional product review systems. Additionally, EthReview effectively mitigates the aforementioned limitations of centralized product review systems. Furthermore, it eliminates the need for a central authority or trusted third-party for validating the integrity of the posted reviews. The major *contributions of this paper* are summarized as follows:

- We present EthReview, which is an Ethereum blockchain-based system that employs a P2P consortium network of randomized endorser nodes for validation and verification of the reviews.
- We propose a two-token system. The *Product Review Authorization Token (PRAT)* is used for review authorization and the *Product Review Discount Token (PRDT)* is used for rewarding the honest reviewers by giving them a discount on products and elevating their status to endorser nodes.
- We also present a thorough security and performance analysis of the proposed system to evaluate its resiliency against various rating frauds.

The rest of the paper is organized as follows. In Section 2, the background information on Ethereum blockchain and ERC20 tokens are presented. A literature review of existing blockchain-based review systems is presented in Section 3. Section 4, presents the proposed solution along with its all major components and algorithms in detail. In Section 5, the proposed solution is analyzed and evaluated for both system robustness and security. Section 6 discusses the limitation of proposed solution. Finally, we conclude the paper in Section 7.

2. Ethereum and ERC20 Tokens

In this section, we describe the necessary background information related to Ethereum Blockchain and ERC20 Tokens. Ethereum is a public open-source programmable blockchain platform for developing decentralized applications (dApp) [23]. Unlike the traditional blockchains, Ethereum is capable of facilitating programming and flexibility through the use of *Smart Contracts*. Smart contracts are an autonomous and self-executing piece of code[7]. Ethereum is considered more powerful since it allows users to define their own set of procedures instead of following predefined transaction procedures, like in the case of Bitcoin.

In Ethereum, smart contracts are commonly written in *Solidity*[8], which is an Ethereum development language [25]. They are executed within the Ethereum Virtual Machines (EVM) on various nodes of the network. Smart contracts are helpful in verifying enforcement of the predefined rules of conduct and business logic [26]. Transactions that persist in the Ethereum blockchain are digitally signed messages using the *public-private* keys of involved parties. Every transaction on Ethereum is associated with an action and addresses of the sender and receiver. The addresses are the public key(s) belonging to a specific user in the Ethereum network. Ether (ETH) is the internal cryptocurrency traded on Ethereum.

The participating nodes of the network must reach a consensus on the output after every execution of a smart contract. This is important to ensure that the overall state of the machine remains stable and error-proof [23]. Ethereum uses the Proof-of-Work (PoW) consensus protocol[9]. PoW is also used for awarding the nodes in Ethers for verifying transactions (mining) on the Ethereum network.

An important metric to judge the performance of an Ethereum-based dApp is to measure the amount of computational effort required to execute every single operation. This effort can be measured in terms of *gas cost*. The amount paid to nodes on the network for executing requested operations by different transactions is known as the *gas cost*. Calculation of gas cost depends on the number of instructions to be executed every transaction [27]. To prevent too much gas consumption, the creator of the transaction can set *Gas Limit*. In Ethereum, the gas cost is paid by the initiator of the transaction [28].

The process of turning things into digital assets is known as *tokenization*. On Ethereum, tokens are implemented using smart contracts. Subject to the purpose of the creation of a dApp, tokens can be used for various purposes e.g. used as an internal currency, proof of ownership, utility

[5] Promoting the products by injecting fake positive reviews

[6] demoting the products by injecting fake negative reviews

[7] The concept was first introduced in 1994 by Nick Szabo [24]

[8] `https://solidity.readthedocs.io/en/v0.5.3/`

[9] The Ethereum team is pursuing to switch Proof-of-Stake (PoS) protocol. While the PoW protocol demands high computing power, the PoS protocol depends on a node's stake in the system (typically, the amount of currency a node possesses) [20]

tokens, security checks, etc. Ethereum Request for Comment (ERC) provides Ethereum developers with a set of standard tools for building smart contracts that implement the functionality of tokens [29]. ERC20 is one of the many standards that can be used for implementing custom tokens. The majority of the tokens issued on Ethereum are ERC20 complaint. We have followed the same standard of tokenization in our system.

3. Related Work

Various studies emphasize the need for privacy-preserving decentralized review systems in the e-commerce marketplace. For instance, Tadelis enlists the different factors in traditional review systems that cause bias in the buyer/seller's feedback in [30]. It has been demonstrated through research and experiments that the user-generated feedback systems are often prone to bias and influence by the sellers. Fear of retaliation by the seller in case of giving negative feedback and the importance of selecting the type of feedback system to be deployed in the online marketplace are the factors that have a major contribution to biasness in feedback systems. Different ways to control such bias have been presented in the work of Cai and Zhu [18]. Additionally, this study also discusses several possible attacks on user-driven review systems, e.g., whitewashing, constant, collusion, and Sybil attacks. In review systems, these attacks are often performed in pursuit of ballot-stuffing or bad-mouthing.

Various approaches have been presented in the literature for mitigation of the aforementioned attacks on the review systems with the aid of security protocols, machine learning, and blockchain technology. We have classified the existing proposed solutions into two main categories-incentive-based or non-incentive-based.

3.1. Non-Incentive-Based Models

In non-incentive based models, the sellers or buyers are not rewarded for posting or validating the reviews. A blockchain-based reputation management system to address the issues regarding fake reviews is proposed in [31]. The framework has two components: generating and accessing the review blockchain. Based on the textual content, it analyzes the reviews for being fake using the IBM Watson Analytics[10]. The sellers, in this proposed model, are responsible for sending the review request. This can result in malicious activity, such as, if the seller decides not to send a review request to the buyer. In this case, the buyer will be deprived of posting reviews. Similarly, the seller can also perform a ballot-stuffing attack. The reviews in this proposed model have an expiry time associated with them and gradually get deleted if the expiry is reached. Additionally, this study does not discuss the security issues and raises questions about the resilience it can provide against various rating attacks.

A P2P rolling blockchain reputation system is presented in [32], an extension of [33]. The proposed system has two main goals. Firstly, to defy some of the most common rating attacks faced by the review systems such as the Sybil attack, collusion attacks, re-entry attack, and 51% attack. The second goal is to contribute a generalized reputation system that can be incorporated into any network. In the proposed system, the reputation score is controlled by the client. Client-controlled means that reputation score is calculated using the parameters set by the clients. This can contribute to biasness since the parameters selected will differ from person to person. However, to overcome the scalability issue of blockchain, there are some negative impacts induced by this proposed solution. The increased demand on the miners for resources to store blockchain would have an adverse effect of consequently leading towards a decreased number of miners in the system. The decrease in the number of miners would in return hurt the overall security of the system. Also, since the nodes have to request data from a pool of miners this will increase the time for a node to calculate a user's reputation. Moreover, delays would incur due to the network latency and processing of the requests by the miners. Furthermore, the proposed solution has not been implemented and is not resistant to intelligent colluding attacks.

A privacy-preserving decentralized reputation system is proposed in [34]. The main goal of the proposed solution is privacy-preservation and well-formedness. To attain its objectives, the model utilizes homomorphic encryption and non-interactive zero-knowledge proofs. Privbox model ensures three characteristics: (1) guarantees the privacy of the consumers without depending on any central authority; (2) ensures that the consumer feedback ratings abide by the range imposed by the system; (3) permits the service providers and consumers to verify the computed reputation scores without the intermediation of a central authority. Implementation of the PrivBox system demonstrates that to attain decentralization and privacy preservation the proposed system has computation and communication overheads. Also, its key focus is on the privacy of the consumers more than the quality and tampering of the reviews.

In [28], a reputation system for the Peer-to-Peer (P2P) marketplace is proposed. The model aims to verify the quality of data traded before a transaction occurs based on the reputation of the data seller. The core element of this system is the Ethereum Smart Contract which replaces central authority and performs the business logic and rules. In this model, an increase in the size of the review string effects the gas amount required for a successful transaction. For example, the default gas limit is 90,000 Wei (unit of Ethereum's internal currency) to write a review of 32 characters. A gas limit of 150,000 Wei will be charged to write a review of up to 96 characters. A prototype of the proposed solution is implemented but no

[10] https://www.ibm.com/watson-analytics

solution is suggested to prevent different rating attacks. Also, an overall reputation score is not generated against each data owner. The amount of gas consumed in a transaction is determined by the input value and the number of lines of code to process. Additionally, in this model, the reviewer has to pay for the gas cost of posting the review, i.e., the reviews are expensive and hence can demotivate the buyer from sending feedback.

Table 1: Comparison Table of Related Work and EthReview

Paper	Underlying Architecture	Model Type	Main Focus	Implemen-tation	Attacks Prevented				Gas Cost Consum-ption Analysis
					Sybil Attack	Ballot Stuffing	Bad Mouthing	Collusion Attack	
[34]	Bitcoin	Non-incentive	Review Integrity	✗	✗	✗	✗	✗	N/A
[35]	Variation of Bitcoin Protocol	Non-incentive	Establishing P2P reputation score by overcoming scalability issues of Bitcoin	✗	✓	✗	✗	✓	N/A
[37]	Homomorphic Cryptographic Non-Interactive Zero-Knowledge Proofs	Non-incentive	Preserving privacy of reviewers in decentralized environment	✓	✗	✗	✓	✗	N/A
[29]	Ethereum Smart Contracts	Non-incentive	Review system that can confirm the reputation of a data owner or the data traded in a P2P marketplace	✗	✗	✗	✗	✓	✗
[38]	Bitcoin Vouchers are used to give incentives	Incentive-based	Controlling feedback abuse	✗	≈	✗	✓	✗	N/A
[39]	Blind Signatures	Incentive-based	Usage of internal tokens for review authorization and incentive distribution	✗	✓	✗	✗	✗	N/A
[12]	Ethereum Smart Contracts Tokens	Incentive-based	Automation of incentive distribution IPFS to store accepted reviews	✓	✗	≈	≈	✗	✗
EthReview	Ethereum Smart Contracts ERC20 Tokens P2P Consortium Network	Incentive-based	Maintain Tamper-Proof reviews Resilience against Rating Frauds Validate and Verify Reviews	✓	✓	✓	✓	✓	✓

✓: Done ✗: Not Done ≈: Partially Mitigated

3.2. Incentive-Based Models

Another problem with the consumer feedback systems from a buyer's perspective is the lack of motivation for providing reviews of the product they have purchased and used. To overcome this, some e-commerce platforms such as Amazon have started to incentivize the buyers for their valuable feedback [35].

Carboni has demonstrated how an incentive-based feedback model can be implemented on top of the Bitcoin protocol in [36]. The proposed approach uses co-signed vouchers for giving incentives to the buyers for leaving feedback about the purchased service/product. Every voucher constitutes a voting fee and an incentive amount. The incentive can only be utilized if both the consumer and producer co-sign the voucher. A voting fee is 3% of the total payment paid by the consumer. If the voucher is co-signed then the reputation of the producer is incremented by the voting fee otherwise no change is reflected. In this proposed model, expensive products could gain a high reputation very quickly than low-price products. Moreover, the model is not resistant to Sybil attacks. It is possible for a dishonest seller to easily create new identities and pump his reputation score. Additionally, negative feedback is not accepted by this model. A consumer may leave positive feedback just for the incentive even if they are not satisfied with the service. It is also possible for a consumer to take the incentive and send negative feedback in response, but since the voucher has already been co-signed, the incentive cannot be reverted. Furthermore, the proposed model is not collusion resistant. Neither can the performance of this system be analyzed since it has not been implemented.

An incentive-driven and trustless blockchain-based decentralized privacy-preserving reputation system has been proposed in [37]. This model offers incentives to the participants (sellers/buyers) for mining and validating the review blocks using the Proof-of-Stake (PoS) consensus algorithm. The system uses internal coins and tokens for this purpose. Tokens are used as the proof-of-transaction in the system to authorize a customer to post a rating. Blind Signatures[11] are used to ensure the anonymity of the customers. The exchange of tokens helps in limiting ballot-stuffing attack as the sellers can not create as many tokens as they desire. Authorization tokens that guarantee acceptance of review are transferred to the buyer before the transaction takes place. Consequently, this allows fake reviews to be accepted by the system. Moreover, it is necessary to determine a mechanism that would provide enough coins to the sellers to provide adequate tokens to their customers and at the same time would limit the ballot-stuffing attack. There can be two ways in which the customer can receive a blinded token from the sellers - the seller receives money from the buyer first and then delivers a token or vice versa. However, both of these solutions are prone to malicious behaviors. In the first case, the seller could first receive money from the customer and then refuse to send him a blinded token. Therefore, leaving the customer deprived of his right to issue ratings. In the second case, if the customer first gets the blinded token and then has to pay the seller, the customer could abort the transaction without sending the money. Even though no transaction has taken place here, the customer still has the chance to use the token and issue a review. Since these transactions cannot be verified, this would result in an opportunity for the adversary to perform Bad-mouthing and Ballot-stuffing attacks against the seller. Furthermore, the protocol has neither been implemented nor tested and hence its efficiency and performance cannot be guaranteed.

Salah et al. have proposed an Ethereum-based smart contract for the automation of the process of distributing incentives to the reviewers in [12]. The key aspect of this paper is the use of the Inter-Planetary File System (IPFS)[12] with blockchain to create a decentralized system and targets the aspect of storage and tracking of approved reviews. Along with this, it uses tokens to reward and allow reviews to be uploaded. Since the sellers upload the rules of conduct, they can be dishonest and allow the Ballot-stuffing attack to happen. Also, the reward amount is set by the seller in this proposed solution giving rise to the opportunity of setting the amount to a minimal value or zero. Moreover, the proposed framework does not have the product purchase and review integrated closely. Since the reviews are not being validated for their textual content this could result in fake content being uploaded resulting in rating fraud. The criteria for review verification is nowhere specified in the paper. The proposed model only validates whether or not the review has come from an Ethereum address that was offered a token from the seller. The rewarding criteria is also very vague since the reviewers will be rewarded for posting anything onto the IPFS server. Even though the tokens are linked to the Ethereum addresses, this could result in a Sybil attack since the address generation is very cheap in a blockchain setup and can allow both Ballot-stuffing and Bad-mouthing attacks. Furthermore, gas cost payment is not managed in this proposed solution. The author has further not provided any security analysis for the proposed framework against the different rating frauds.

It can be observed that the solutions provided so far focus on different properties of the review systems like consumer privacy, automated incentive distribution, etc. There is a need to provide a working system that can integrate multiple solutions. A comparative analysis of the existing related work and EthReview is presented in Table 1. The comparison is based upon the numbers of attacks mitigated, implementation, and performance analysis in terms of gas cost consumption.

[11] https://en.wikipedia.org/wiki/Blind_signature

[12] https://ipfs.io/

4. Proposed EthReview System

In this paper, we propose an online product review system, EthReview. Our proposed solution is based on Ethereum blockchain and aims at mitigation of rating frauds possible against traditional product review systems. An overview of the key components of the proposed solution is presented in Figure 1. In EthReview, product purchase is integrated with the product reviewing process. By this we mean, the system only grants review authorization to a buyer if he has obtained the review authorization token after the product purchase. Furthermore, a consortium peer-to-peer (P2P) network of endorser nodes is proposed for validation of the product reviews. Incentives in the form of discount tokens are also distributed for encouraging honest behavior in the system.

In Section 4.1, we set the foundation by defining the participants involved in our system EthReview. Furthermore, in Section 4.2, we discuss the operations performed by these participants in EthReview. Section 4.3, illustrates the endorsers functionality and how endorsers are selected in EthReview. Section 4.4, explains how the initial endorsement happens in EthReview, followed by the role of endorsers for verifying authentic product refund in Section 4.5. In Section 4.6, we discuss the privacy of participants in EthReview.

4.1. System Participants

There are four main participants in our systems: *Sellers (S)*, *Buyers (B)*, *Reviewers (R)* and *Endorsers (E)*.

1. Sellers are the vendors selling products.
2. Buyers are the purchasers of the products sold over the e-commerce platform.
3. Reviewers are a subset of the buyers. These are participants who submit reviews about purchased products.
4. Endorsers are the selected subset of reviewers who behave honestly by posting truthful reviews about the products. The process of endorser selection is automated and randomized by the use of smart contracts. These endorsers nodes are responsible for validating the credibility of the posted reviews by endorsing them (i.e. cast an upvote/downvote against the review). Endorsers endorse the product reviews based on their recent purchase experience about the products. Further details are explained in Section 4.3.

The different operations that the aforementioned participants can perform within the system are described in Section 4.2 below.

4.2. Operations

Any operation performed by the participants in our Ethereum-based product review system will incur *gas cost* in terms of Wei[13]. Gas cost is required for running computations and execution of smart contracts and is essential to Ethereum blockchain.Before performing any operations in EthReview, *Sellers (S)* and *Buyers (B)* must register themselves with their credit/debit card numbers. Only one registration is accommodated per unique credit/debit card number. Upon registration, every user is allocated a unique *userID* in the system which is mapped to their Ethereum address(es) and is used for tracing all the activities of an individual user in EthReview.

Sellers (S) are allowed to *Add* new products, *Remove* or *Update* their existing products. A seller (S) adds a product (P) in the system after successful deduction of gas cost ($GCAddP$) incurred for adding product on Ethereum blockchain from his *Account(S)*. If the seller (S) does not have enough amount of Ether/Wei in his *Account(S)*, the transaction for adding the product would fail. Moreover, for listing a *product (P)* on EthReview, an additional amount of one Ether is deducted from seller's account, *Account (S)*. This amount is deposited into the review fund, *ReviewFund (P)*, reserved for that particular product. This reserved fund will be used for paying the gas cost incurred to the *Buyer (B)* for reviewing this product. Leveraging the gas cost incurred for review is important to make the reviewing process less expensive for the buyers and to motivate them for submitting reviews. Hence, for this proposed solution, we believe that a better economical design choice is to shift the gas cost of reviewing products on to the sellers. In case of successful listing of a *product (P)*, the balance in the seller's account and *ReviewFund (P)* will reflect the change shown in equations 1 and 2.

$$Account(S) = Account(S) - [GCAddP + 1Ether] \quad (1)$$

$$ReviewFund(P) = ReviewFund(P) + 1Ether \quad (2)$$

Buyers (B) are only allowed to *Buy* products and *publish reviews* about the purchased products. Once a product (P) has been listed, a buyer (B) can purchase the product using their *Account(B)*. A buyer must possess an account balance greater than the product cost (*pCost*) to buy the product. On purchase of a product, gas cost (*GCBuyP*) along with product cost (*pCost*) is debited from buyer's *Account(B)*. An ERC20 token, known as Product Review Authorization Token (*PRAT*), is then transferred to the buyer's Account(B), increasing the *TokenBalance(B)* by one. The product cost (*pCost*) is credited to seller's *Account(S)*, while the gas cost (*GCBuyP*) is consumed by Ethereum network for processing the transaction and smart contract computations. On the successful purchase of a product, the accounts of the seller and buyer will reflect the changes shown in equations 3, and 4.

$$Account(B) = Account(B) - [GCBuyP + pCost] \quad (3)$$

[13] Wei is the smallest denomination of ether, the cryptocurrency token on the Ethereum network.

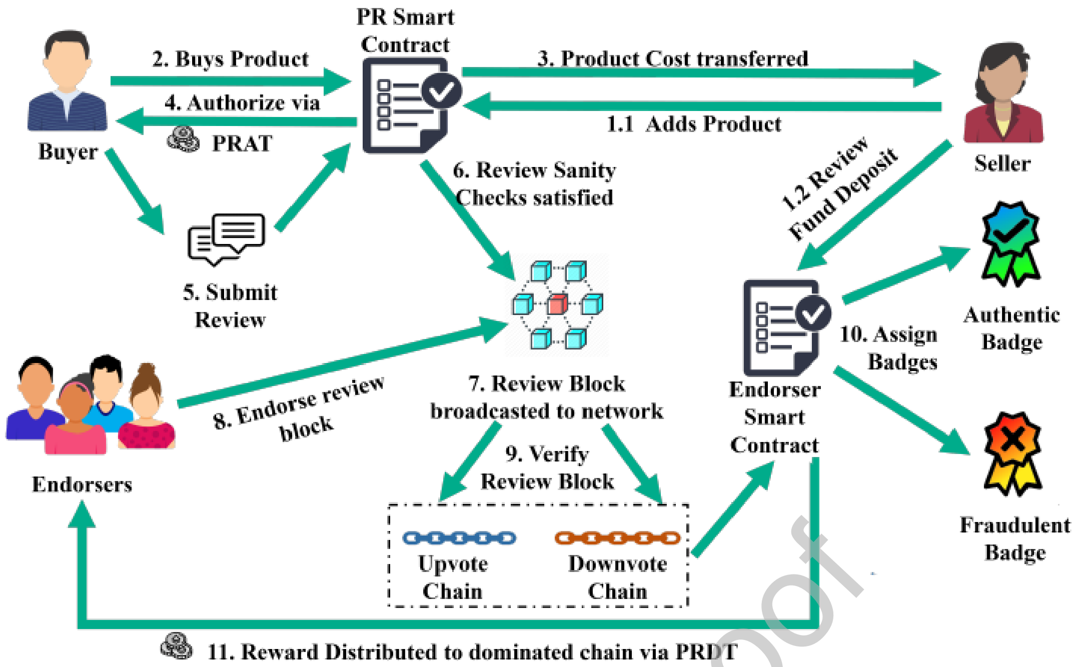


Figure 1: The system overview highlighting key components of EthReview that utilizes different *Smart Contracts* for providing functionalities of product purchase, review and endorsement.

$$PRATBalance(B) = PRATBalance(B) + 1 \tag{4}$$

$$Account(S) = Account(S) + pCost \tag{5}$$

In order to encourage the buyers to post reviews, it is important to make the posting of reviews free of cost. Hence, PRAT tokens are used for paying the gas cost for posting reviews in our system. The PRAT tokens are fueled using the *ReviewFund* deposited by the product vendors. This review fund is maintained by the *Endorser Smart Contract*. For every one Ether deposited by the seller for a particular product at the time of product listing, hundred PRAT tokens are reserved in *ReviewFund*. This means that for every one Ether deposited, hundred reviews of the product can be facilitated. These PRAT tokens are used for compensating the gas cost of publishing these hundred reviews. Once all hundred PRAT tokens of the product are consumed, the sellers are asked to deposit additional Ethers to generate new PRAT tokens to facilitate product reviews of every product purchase. In case a seller refuses to deposit Ethers into the *ReviewFund*, no reviews for that particular product would be accepted.

Furthermore, PRAT tokens are used as proof of authorization for reviewing any particular product. In EthReview, PRAT tokens are disbursed only upon the successful purchase of a product. These tokens are bound to products using the product's SkuID and cannot be used to review any other product on the system. Moreover, PRAT expires after 3 days of allocation and becomes invalid for use. However, if the allocated PRAT token is expired and not used by the buyer for reviewing, then in that case the PRAT token is reversed back in the review fund and reflected in the cumulative PRAT token count. This reversed PRAT token can be issued to future buyers for review. Once a review gets posted by the buyer, the PRAT token is *burned* (i.e., destroyed) to avoid the reuse or transfer of token to any other non-buyers.

Algorithm 1, describes the consumption of PRAT token for the reviewing process. In EthReview, a product needs to exist in the system at the time of its review. Furthermore, at the time of product review, it is checked that the reviewer has the possession of an unexpired *PRAT* token through the valid purchase of the product. If all of these constraints satisfy, then the reviewer can successfully review the product and the review gets published on the blockchain for endorser nodes to validate.

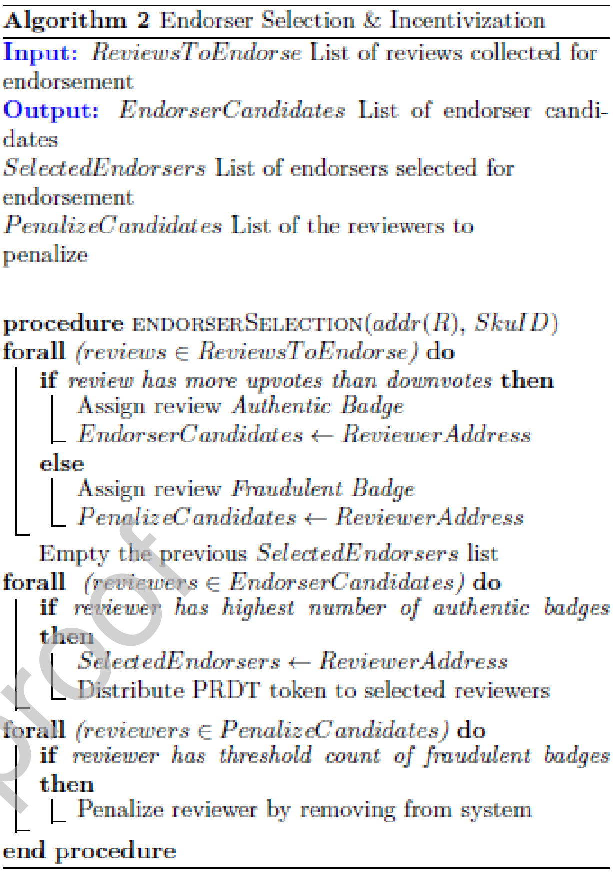
```
Algorithm 1 Review Product
Input: $SkuID$ is unique ID of the product to review
       $userID$ is unique ID of the reviewer
Output: Product review

procedure ReviewProduct($SkuID$)
if (user purchased the product with $SkuID$
    AND has unexpired PRAT token) then
    Allow user to add review
    Add user's review to blockchain
    Add $userID$ to the list of reviewers of the product
    Use PRAT token to pay for gas cost
    Burn the token
else
    Not Allowed to Review the Product
end procedure
```

4.3. P2P Randomized Consortium Network of Endorsers

Every review published in EthReview is endorsed by the endorser nodes based on their personal experience about the respective product. In EthReview, these endorsements, in the form of upvotes/downvotes, are marked against the authenticity of the posted reviews for a particular product. If according to an endorser's opinion after usage of the product, the review is valid and isn't falsely injected to promote or demote the product, they will cast an upvote against the review. An endorser will downvote a review if it is the other way around. The process of selecting these endorser nodes from the reviewers of products is illustrated in 2. To reflect the authentic and fraudulent reviews in EthReview, the product reviews are assigned badges (i.e., authentic/fraudulent) after successful endorsement.

- Authentic Badge: If the review is in synchronization with the cumulative endorsers' experience of the product, then it is given an authentic badge. If a review has greater upvotes, this means that the review is not falsely injected to promote or demote a product. Authentic badges can be assigned to both a positive or a negative review.

- Fraudulent Badge: If the review does not match with the cumulative experience of endorsers with the product, then it is given a fraudulent badge. If a review has greater downvotes, this means that the review is falsely injected to affect the reputation of the product. The fraudulent badge can be assigned to both positive or negative reviews.



Algorithm 2 Endorser Selection & Incentivization

Input: *ReviewsToEndorse* List of reviews collected for endorsement

Output: *EndorserCandidates* List of endorser candidates
SelectedEndorsers List of endorsers selected for endorsement
PenalizeCandidates List of the reviewers to penalize

```
procedure EndorserSelection($addr(R)$, $SkuID$)
forall ($reviews \in ReviewsToEndorse$) do
    if review has more upvotes than downvotes then
        Assign review Authentic Badge
        $EndorserCandidates \leftarrow ReviewerAddress$
    else
        Assign review Fraudulent Badge
        $PenalizeCandidates \leftarrow ReviewerAddress$
    Empty the previous $SelectedEndorsers$ list
forall ($reviewers \in EndorserCandidates$) do
    if reviewer has highest number of authentic badges then
        $SelectedEndorsers \leftarrow ReviewerAddress$
        Distribute PRDT token to selected reviewers
forall ($reviewers \in PenalizeCandidates$) do
    if reviewer has threshold count of fraudulent badges then
        Penalize reviewer by removing from system
end procedure
```

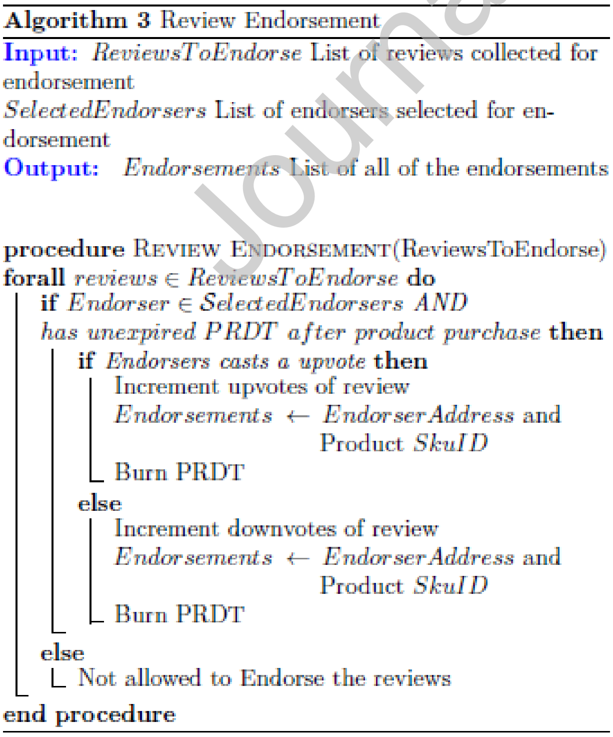
The following are key steps followed in Algorithm 2 that forms the basis of endorsers functionality. These steps are logically implemented in EthReview as *Endorser Smart Contract*.

1. The reviews that receive more upvotes than downvotes from the group of endorser nodes are given *authentic badges*. The reviewers of these reviews are made part of the *EndorserCandidates* list, whereas, the reviews with more downvotes are given *fraudulent badges* [14]. The reviewers of these downvoted reviews become part of the *PenalizeCandidates* list.
2. From the list of *EndorserCandidates*, 60% the reviewers having highest number of *authentic badges* against their reviews overall in the system are selected and made part of the *SelectedEndorser* list. This *SelectedEndorser* list will perform endorsement for the next round of product reviews for which they are awarded Product Review Discount Token (PRDT).
3. From the pool of *PenalizeCandidates*, the reviewers having the threshold count of *fraudulent badges* overall in the system against their reviews get blacklisted and are removed from the system.

[14] *Badge* is a boolean variable that can be true (authentic badge) in the case when upvotes of a review are greater than its downvotes. It would be false (fraudulent badge) when downvotes are greater than the upvotes.

4. All the reviewers in the *SelectedEndorser* list are awarded (*PRDT*) tokens by the smart contract as a reward for honest behavior. PRDT is a system token that is associated with a particular product using it's *SkuID*. Using the PRDT, the promoted endorsers in *SelectedEndorsers* list can purchase the associated product at a discounted price as a reward. The endorsers can endorse multiple un-endorsed reviews of the particular products for which they are awarded PRDT tokens. The endorsers are allowed to endorse each un-endorsed review of that particular product only once.
5. After a complete round of endorsement, the *SelectedEndorser* list is re-initialized with a new list of *EndorserCandidates*, following the same procedure from Step 1.

The review endorsement mechanism by the reviewers augmented to endorsers in *SelectedEndorsers* list is illustrated in Algorithm 3. For an endorser to upvote/downvote a review, they must belong to the *SelectedEndorser* list, should possess an unexpired PRDT token for the product associated with the review to be endorsed. If all of the aforementioned sanity checks are satisfied, then the endorser is allowed to endorse the reviews of a particular product. Once an endorser casts an upvote/downvote against a review, the *Endorser's Address* and *Product SkuID* is added to the list of *Endorsements* to make sure that the endorser only endorses an individual review only once. The PRDT token allocated to the endorser is then burned (i.e., destroyed).



```
Algorithm 3 Review Endorsement
Input: ReviewsToEndorse List of reviews collected for endorsement
SelectedEndorsers List of endorsers selected for endorsement
Output: Endorsements List of all of the endorsements

procedure Review Endorsement(ReviewsToEndorse)
forall reviews ∈ ReviewsToEndorse do
    if Endorser ∈ SelectedEndorsers AND
    has unexpired PRDT after product purchase then
        if Endorsers casts a upvote then
            Increment upvotes of review
            Endorsements ← EndorserAddress and
                           Product SkuID
            Burn PRDT
        else
            Increment downvotes of review
            Endorsements ← EndorserAddress and
                           Product SkuID
            Burn PRDT
    else
        Not allowed to Endorse the reviews
end procedure
```

4.4. Bootstrapping Endorser Nodes

To facilitate the initial endorsement (upvotes/downvotes) of reviews in EthReview, the *Endorser Smart Contract* will randomly choose a threshold count of reviewers from the first batch of received product reviews. These selected endorsers will then be given PRDT tokens as an incentive for performing endorsements. These PRDT tokens are bound with specific products using product SkuId. After the allocation of PRDT tokens, these reviewers (now acting as endorsers) can use the PRDT tokens for buying the product bound against PRDT and endorsing the reviews of that product. For the next round of endorsement, a new set of endorser nodes will be selected as per the mechanism explained in Section 4.3.

4.5. Endorsers and Refund Process

Refunds are a common activity in an e-commerce system. Refunds are usually claimed when buyers are not satisfied with the quality of the product received. It is important to be sure that the buyer is claiming a valid refund and the product is indeed bad. This is necessary to avoid fake refunds targeted towards to demotion of a product. Therefore, whenever a refund request from a buyer is made in EthReview a threshold number of previous endorsers of the product will be selected randomly for processing the validity of the refund claim. These endorsers have previously endorsed the reviews of the product under claim and have used the specific product. Based on the cumulative judgment of these endorser nodes, the refund would be approved or disapproved.

4.6. Privacy of the participants:

In this section, we discuss the privacy of the participants in our proposed model. Our proposed model ensures that the identities of the participants are not revealed to one another. In the Ethereum blockchain setup, every participant uses a unique public-private key-pair for addressing. Public keys are used to create Ethereum addresses. Whereas, the private keys are used for signing Ethereum transactions for sending and receiving Ether(s). In our system, every participant has an Ethereum address(es) associated with their account. While performing any operations in the system, the identities of participants remain anonymous through Ethereum addresses. This mechanism maintains the privacy of participants in our system.

5. Analysis and Evaluation of EthReview

In this section, we first present the adversary model in Section 5.1, by discussing different rater categories in EthReview and the rating frauds associated with them. Then, we evaluate the resilience of EthReview in the light of this adversary model in Section 5.2. Lastly, we evaluate the performance of our proposed solution, EthReview in terms of the gas cost needed to perform transactions and smart contracts analysis, in Section 5.3.

Table 1: Rating attacks possible by different rater categories and system defense mechanism

Rater Category	Attack Scenarios	EthReview Defense
Happy Honest	No attacks since reviews are based on honest positive buying experience.	—
Unhappy Honest	Buyer and seller collusion inside system. Central authority and seller collusion	Data published onto blockchain is immutable Hence, reviews cannot be altered or removed.
Happy Dishonest	Ballot-stuffing coupled with Constant, Whitewashing and Colluding Attacks through exploiting Sybil attack.	One review per buyer with authorization token. Reviews validation by P2P endorser nodes. Creation of identities is hard. Awarding random PRDT tokens to endorsers.
Unhappy Dishonest	Bad-mouthing coupled with Constant, Whitewashing and Colluding Attacks through exploiting Sybil attack. False Refund Claims	One review per buyer with authorization token. Expensive for adversary Reviews validation by P2P endorser nodes. Creation of identities is hard. Awarding random PRDT tokens to endorsers.

5.1. *Adversary Model*

In this section we construct the adversary model by first identifying different kinds of rating frauds conducted by adversaries in EthReview. Secondly, we identify the possible types of rater categories that are involved in EthReview. Thirdly, we proceed by associating the investigated rating frauds with the identified rater categories.

5.1.1. *Rating Frauds*

This section discusses the various kinds of rating frauds that are prevalent in EthReview. Rating frauds are generally classified into two main categories: *Bad-mouthing* and *Ballot-stuffing* [19]. Most prevalent attacks that are performed in conjunction with ballot-stuffing or bad-mouthing, as described by [18, 38, 4, 8, 9], are enlisted below.

1. *Collusion Attacks*: Two or more entities of the system can collude together to perform rating frauds. In our proposed system, the objective to conduct colluding attacks can be either of the following:
   - Collusion of buyers and sellers inside the system boundary to modify/remove an honest negative review in return of some incentive e.g. if a buyer after the purchase of the product honestly does not likes it and posts a negative review. The seller can collude with the buyer by bribing him to remove/alter his negative review in return for some discount voucher or any other form of incentive.
   - Collusion of sellers and buyers outside the system boundary to promote or demote products e.g. a seller can hire some fake buyers outside the system to buy products in the system and give fake positive or negative reviews about them.
   - Collusion of the central authority and the sellers to promote or demote products, to debar certain buyers from reviewing, to post only selective reviews, alter the reviews, etc.
2. *Constant Attacks*: The aim of the adversary here is to target specific products and constantly bombard them with fake positive or negative reviews to demote or promote them.
3. *Whitewashing Attacks*: This attack constitutes a fraudulent rater posting fake reviews about products, then exiting the system whitewashing his identity, i.e., re-enter with a new identity. Due to new identity, it becomes hard to detect whether or not the same reviewer was involved in opinion spamming earlier.
4. *Sybil Attacks*: It is intended to make the attack spectrum wider by creating multiple identities in a system and performing any of the aforementioned attacks. For example, fraudulent raters can generate multiple identities and through collusion attacks inject fake reviews to achieve ballot-stuffing or bad-mouthing. Sybil attack can be exploited to perform any of the aforementioned rating frauds (collusion, constant, whitewashing).
5. *51 % Majority Attack* : 51% attack refers to the adversary controlling more than 51% of the nodes in the blockchain network. The attackers would be able to prevent confirmations of new transactions, halt payments between some or all users, double-spend coins, etc. Similar to any blockchain application, this attack is possible in the context of EthReview when the adversary nodes manage to become more than 51% of the endorser nodes, thereby controlling the validation process of reviews through majority influence.

5.1.2. Association of rater categories with rating frauds

A bad review can exist due to two possibilities: either the product is bad in reality or the review is intended to disturb the credibility of the respective seller of the product in the market. Therefore, it is imperative to identify different categories of raters that would be involved in the product review process. We have divided the raters into four main categories based on his intent of review:

1. *Happy Honest*: This is our ideal rater, the rater who likes the product and gives honest positive feedback about the product.
   Rating Frauds: These ideal raters do not pose any rating frauds in the system. These raters are of utmost importance to the product sellers since they help in developing trust about the products in an e-commerce marketplace. Also, the reviews from this kind of raters have a positive impact on the sales of the product.
2. *Unhappy Honest*: This too falls in the category of ideal rater because he is the rater who does not like a product and gives an honest bad review about it.

   Rating Frauds: Negative reviews affect sales growth and gained profit. Therefore, the collusion attacks of sellers with buyers inside the system and collusion of central authority with the sellers can happen in this case to remove negative reviews. For example, a seller may ask an unhappy honest rater to remove or alter his negative review in return for some incentive in the form of discounts, complimentary items, etc. Moreover, sellers could collude with the central authority to suppress negative ratings by displaying all positive ratings on top. Furthermore, some central systems only accept positive ratings for its products [18].
3. *Happy Dishonest*: These are one of the main categories of fraudulent raters that are critical yet important to handle. The happy dishonest rater is the fraudulent rater who tends to promote a product by giving fake positive reviews (Ballot-stuffing). This rater intends to inclusively augment positive ratings to a product. This type of rater can be the seller himself forging the buyer identity or fake buyers hired by the seller.

   Rating Frauds: This rater category can pose all the discussed attacks (including collusion, constant and whitewashing attacks) by exploiting the Sybil attack and creating multiple fake identities in the system.
4. *Unhappy Dishonest*: This is the other main category of the fraudulent raters who want to demote a product by injecting fake negative reviews (Bad-mouthing). The intent of this rater, unlike happy dishonest, is to inject fake negative reviews to demote his competitor's product. This type of rater can be the seller himself or fake buyers like in the case of the happy dishonest rater.

   Rating Frauds: All types of attacks (including collusion, constant and whitewashing attacks) exploiting Sybil attack can be performed by both unhappy dishonest and happy dishonest rater with the difference of the rater's intent. Happy dishonest cause these attacks to augment product ratings while the motive of unhappy dishonest is to reduce the ratings.

5.2. Security Analysis of EthReview against Adversary Model

In this section, we analyze the defense mechanism of our system against the adversary model, described above in Section 5.1. Table 1 provides a summary of the different rating attacks possible by each rater category and the EthReview defense mechanism to mitigate them.

EthReview is resilient against rating frauds by integrating product purchases and authorization tokens with product reviews. This allows the buyers to only review the products that they have purchased. By decentralizing the system, we have eliminated the need for a central party which removes the chances of a third party tampering with the posted reviews. Once the review is published in EthReview, it remains there forever and cannot be tampered. EthReview accepts both negative and positive reviews. These reviews are then validated by the endorser nodes and assigned authentic and fraudulent badges that help in evaluating the truthfulness of the reviews in the system.

Below we separately discuss the resilience of EthReview against different attacks possible in the adversary model, as discussed in Section 5.1.1.

1. *Collusion Attacks*: EthReview prevents collusion of buyers and sellers inside the system or seller's collusion with central authority by harnessing blockchain as an underlying technology. Our system isn't controlled by any single authority which can intervene in the review process to only show positive reviews or modify/delete the reviews once posted. Moreover, the tamper-proof properties of blockchain doesn't allow the buyers to change the review once it is posted.
   The collusion of sellers with multiple fraudulent buyers' identities to inject fake reviews in the system to achieve ballot-stuffing or bad-mouthing is prevented by introducing the P2P multi-node endorser model. Let us assume that a seller introduces multiple fake buyer identities into the system. These identities can be used to purchase the products and inject fake reviews towards them. In the case of bad-mouthing, this attack is much more expensive because the adversary has to invest in product purchases of the target competitor. Moreover, by doing so adversary is going to increase his competitor's revenue and sales as well. Therefore, it is not an economic choice for adversary at the first place. Still if the adversary is capable of paying the product cost, the credibility of

the reviews is still validated by the group of randomly selected endorser nodes. If the fake reviews are not in consensus with the endorser's recent experience of the product, these fake reviews will receive downvotes and fraudulent badges. Furthermore, the reviewers who have injected these fake reviews will get penalized from the system if the threshold amount of fraudulent badges against the reviewer's addresses is reached. Both the cost to invest and the lack of control on endorserâĂŹs selection provide sufficient defense in depth.

In the case of ballot stuffing, the product purchase cost is ultimately reimbursed to the adversary. Therefore, in contrast to bad-mouthing, this attack may seem to cost nothing to the adversary. However, we should note that sellers still have to additionally invest in the review fund for PRAT tokens. Further, the amount consumed for posting the reviews will never be reimbursed either. Both these conditions act as a deterrent to the adversary. Further, if we assume that the seller is capable of bearing the PRAT tokens cost, EthReview requires validation from endorser nodes that may not be in the adversaryâĂŹs control. Hence, the fraudulent reviews are debarred at this first security layer of EthReview. Moreover, assume that even if the adversary manages to become an endorser node by behaving honestly in the system. To mitigate this, PRDT tokens act as another layer of defense. The *Endorser Smart Contract* finds un-endorsed reviews and assigns PRDT tokens to the selected endorsers for the endorsement of these reviews. PRDT tokens are bound to the product SkuID of the un-endorsed reviews. In this way, the endorsers cannot freely endorse any reviews of their choice, but in fact, they are constrained to act on selected reviews for which PRDTs are assigned. We believe that this restriction sufficiently hardens the defense and minimizes any gains for the attacker by the fraudulent promotion of buyers to endorser nodes.

2. *Constant Attacks*: In our model, a reviewer has to buy a product to review it. Also, the reviewer can only review the product using PRAT associated with the product SkuID. In addition to that, each review of a particular product can only be endorsed by an individual endorser once. All these system strategies help in eliminating targeted product entity attacks in the system.

3. *Whitewashing Attacks*: In order to join the system, participants have to register on our system using their credit/debit cards. Upon registration on EthReview, it is verified that the record with the same card isn't present in the system before. Therefore, EthReview doesn't allow the participants to leave the system by whitewashing their bad reputation and join the system again with the same identity.

4. *Sybil Attacks*: In EthReview, we limit the creation of fraudulent identities by registering candidates with their unique credit card numbers and *userID*. The *userID* is only issued once at the time of registration. Ethereum addresses in EthReview are linked with this *userID*. Credit card linking with *userID* is added as an extra security layer in EthReview to make the Sybil attack difficult for the adversary. However, it doesn't guarantee the elimination of the Sybil attack completely in EthReview. In the scope of our system, Sybil attacks are exploited to perform any of the other attacks (collusion, constant, whitewashing). EthReview prevents the adversary to exploit the Sybil attack to perform any of the aforementioned attacks as discussed in the respective section of these attacks. However, EthReview doesn't guarantee the complete defense against Sybil attack independently.

5. *51% Majority Attack*: Our system incorporates all the checks in smart contracts to make the review process valid and authentic. We assume that it is very difficult for the adversary to submit fake reviews in the presence of these strict security mechanisms. We have discussed our system defenses against all the adversarial attacks in the previous sections thoroughly. However, if the adversary still manages to bypass every defense layer of EthReview and can control more than 51% of endorser nodes then the system will be compromised like any other blockchain application [39, 40, 41, 42, 43]. To address 51% there are various approaches proposed in the literature that will be incorporated in future work [44, 45].

6. *False Refund Claims*: Unhappy dishonest raters can try to claim refunds by showing fake dissatisfaction about a product to compensate for product cost in case of bad-mouthing. To address this problem in our system, a group of randomized endorser nodes, who have bought the claimed product, is selected by the smart contract. These endorser nodes verify whether the claim of refund is authentic or not.

5.3. Implementation and Performance Analysis

In this section, we evaluate the performance of our smart contract in two dimensions, namely, gas cost consumption and the security analysis of the smart contract for vulnerabilities in the code.

5.3.1. Gas Cost Analysis

For testing purposes, we deployed our prototype on *Ropsten Test Network*[15] that runs the same protocol as

[15] https://ropsten.etherscan.io/

Table 2: Gas costs consumed by different modules of EthReview

Function Caller	Function Name	Gas Limit (Units)	Gas Used (Units)	Gas Price (Gwei)	Total (ETH)	Total (USD)
Seller	Add Product	274663	183109	2.8	0.000513	0.14
Buyer	Buy Product	100557	64785	2.8	0.000181	0.050
Endorser	Endorse Review	107832	87654	2.8	0.000245	0.040

Ethereum. The operations of EthReview are performed with test ethers using *Metamask Wallet* [16] to evaluate the incurred gas cost of these operations. Gas cost is used as a parameter to judge the performance of EthReview in terms of the speed of performing operations. In our system, a call to the module *Add Product* causes gas cost deduction from the seller's account. On the other hand, the gas cost to call *Buy Product* module is deducted from the buyer's account. *Review Product* gas cost is paid through PRAT that is assigned to the buyer for submitting the review. Gas cost for *Endorsing* the review is deducted from the Endorser's account. Usually, smart contracts that change the state of the variables require more computational power and hence more gas is utilized.

Table 2, shows the gas cost in *Ethers* and *USD* that is consumed by each module of our EthReview to perform different operations. As depicted in Table 2, the total gas cost for *Add Product* module is slightly higher than the *Buy Product* module and *Endorse Review* module. This is because the *Add Product* module is changing the state of all the product variables in the smart contract logic of our system. However, as compared to the benefits provided in terms of security, trustworthiness, and speed, the amount incurred in the form of gas cost is negligible and does not affect the overall cost of the system. We have performed our test by specifying the static gas price of 2.8 Gwei that is enough to carry out computations of our smart contract with good speed. For this paper, we could not find any other system build on top of Ethereum to compare our system's gas cost consumption. As a result, we believe that the overall cost of our system is reasonably small and is therefore acceptable to both the sellers and the buyers without becoming a major budget concern.

5.3.2. Smart Contract Analysis

In this paper, we mainly use smart contracts of the Ethereum blockchain for the architecture and implementation of our proposed solution. Therefore, our scope is only limited to the security of blockchain networks in the light of vulnerabilities exposed by the smart contracts. The security of Ethereum blockchain is highly dependent on the security of smart contracts because it constitutes the business logic of the decentralized application (dApp). For Ethereum-based dApps, exploitation of vulnerabilities in the smart contracts can lead to different attacks, like the DAO attack [46]. Analysis of smart contracts is important since once the smart contracts are deployed on the Ethereum blockchain they are not updatable and are visible to all the users of the blockchain network. Due to this, the security holes and vulnerabilities in the smart contract are visible to everyone. Furthermore, in our proposed system, smart contracts are important for maintaining the state of the blockchain. Henceforth, there is a need to analyze the smart contracts before deploying it on the main Ethereum network. This is crucial to avoid any kind of attacks that can be performed by exploiting weaknesses in the smart contracts.

For this purpose, we have used the *SmartCheck* [47] tool. SmartCheck is a static analysis tool for discovering vulnerabilities and other code issues in Ethereum smart contracts written in the Solidity programming language. It analyzes the code for approximately 75 different checks related to 3 vulnerability classes, namely, blockchain, language, and model. For more information on the classification and types of vulnerabilities analyzed, we refer the user to the SmartDec Github repository[17]. Our smart contracts have a total of 350 lines of code which is analyzed by SmartCheck.

Table 3, illustrates the vulnerabilities detected in our smart contracts as a result of SmartCheck analyzer. Vulnerabilities detected included storage access (possibility of overlap attack by accessing the storage slots can change the state variables) and gas limitations (extra gas cost consumption due to checking of state variables as a condition of a loop). Storage access vulnerability was detected because the length of a dynamic array was being changed directly instead of the standard method encoded in the SmartCheck knowledge base. From these results, it is helpful to note that our smart contracts are safe from multiple types of vulnerabilities and exploits.

Table 3: Vulnerabilities detected in EthReview Smart Contracts using SmartCheck

Vulnerability Class	Vulnerability Group	Detection in EthReview Smart Contracts
Blockchain	Block content manipulation	✗
	Contract Interaction	✗
	Gas Limitations	✓
	Message Structure	✗
	Ether transfer	✗
Language	Arithmetic	✗
	Storage access	✓
	Internal control flow	✗
Model	Authorization	✗
	Trust	✗
	Privacy	✗
	Economy	✗

6. Discussion

Product review systems serve a pivotal role in establishing trust between buyers and sellers on an e-commerce

[16] https://metamask.io/

[17] https://github.com/smartdec/classification

platform. Therefore, it is important to build reliable review systems that guarantee tamper-proof reviews and is resilient against rating frauds. In this paper, we have proposed a blockchain-based solution to build robust and reliable review systems. Even though our solution mitigates most of the issues of a centralized review system, it does have some limitations. In this section, we discuss the possible limitations of our proposed solution and how we aim at overcoming them.

1. Extending to multiple reviews per buyer: For the sake of simplicity, we kept one review per buyer in the model. However, our system can be easily modified to entertain multiple reviews by a single reviewer. It can be done so by appending the updated review to the previously posted review, creating a chain of reviews per reviewer. This can easily allow customers to update reviews in case they change their mind after using the product for a while. To control opinion spam, we can limit the number of updates that a reviewer can perform. For example, in one configuration, a reviewer can be given the opportunity to post only two reviews, one as the original and second as its update. Both of these reviews will be maintained by the system in the review chain.
2. Scalability of EthReview: E-commerce platforms handle thousands of transactions per second. It is important to note that though the rate of selling and buying activity is much higher, the review activity is comparably less demanding. Nevertheless, the scalability and performance of review systems hold immense importance. Currently, Ethereum is capable of handling 15 transactions per second [48].
   Assuming a product is already listed on the system, one transaction each is required for buying, reviewing, and âĂŸnâĂŹ number of transactions are required for endorsing the review. Here âĂŸnâĂŹ represents the number of endorsers candidates selected by the Endorser Smart Contract for the endorsement of the respective reviews of the product. This means that for one purchase one additional transaction of review and âĂŸnâĂŹ number of transactions for endorsements are made. At any time âĂIJTâĂİ, if we assume âĂIJn = 3âĂİ, then three sets of transactions (5*3) can be optimally performed per second for a single product in our system. However, three sets of transactions per second are not considered to be the ideal desired outcome in terms of scalability. There are multiple solutions in literature like lightning, sharding, plasma, etc.[49], to scale ethereum applications to a larger number of transactions. . EthReview can easily benefit from these scalability improvement techniques in the same manner as other ethereum applications without any loss of functionality or adding any limitation. . As future work, we aim at experimentally analyzing the scalability and performance of our system in terms of transactions and load testing by applying different scalability techniques for Ethereum dApps to enhance their performance. [50]. The formula for calculating the optimal number of transaction sets that can be performed in EthReview at any time instance âĂIJTâĂİ is given in Equation 6 below:

   $$\frac{15}{B_{tx} + R_{tx} + n} = numberoftransactionsets \tag{6}$$

   In Equation 6, B_{tx} represents the total number of buy transactions, R_tx represents the total number of review transactions, while $'n'$ denotes the variable number of transactions according to the number of endorsers in the system.
3. Integration with existing platforms: With the increase in the trend of e-commerce, there is also an increase in the risks and security breaches on these platforms. We have tested our proposed decentralized application (dApp) on the Ropsten Test Network for gas cost analysis. Ropsten is a testbed for testing Ethereum based dApps before the real-time deployment on the Ethereum network. In this way, we have evaluated the practicality of our system to work effectively. There are two possible ways in which our proposed system can work, either as an independent module or can be integrated as a sub-module of the existing e-commerce system.
4. Review Fund: Currently, in our proposed model, we have proposed a fixed model for the review fund, i.e., the fixed amount of 1 ether is allocated in the review fund for fueling 100 PRAT tokens. However, this is not an economical choice for practical scenarios and is merely for the system's assumption. In future work, we plan the transition to the variable model for deployment purposes. In the variable model, 10% of the product price will be allocated to the review fund. The number of reviews will be fueled from the 10% of the product price in the review fund that will be utilized to issue PRAT tokens for accommodating the product review gas cost price. The variable model will follow the formula depicted in Equation 7 for adjusting the review fund cost for product reviews.

   $$\frac{10\% of product price}{Gas cost of 1 review} = number of PRAT issued \tag{7}$$

7. Conclusion and Future Work

In this paper, we have presented an Ethereum blockchain-based product review system, *EthReview*, to overcome the various rating frauds prevalent in the existing product review systems used on popular e-commerce platforms (like Amazon, eBay, etc.). These rating frauds (ballot-stuffing and bad-mouthing) include augmenting or suppressing the reviews to unfairly promote or demote the products on e-commerce platforms. We have followed a methodological approach to propose a blockchain-based solution that is resilient against these rating frauds.

Firstly, we identified the different rating frauds possible in centralized product review systems. Then we modeled our proposed solution to be resilient against these rating frauds. We made use of the *ERC20 tokens* on the *Ethereum Blockchain* to propose a two-token system for reviewing products and incentivizing honest reviewers. Furthermore, a *peer-to-peer multi-node randomized consortium network* is proposed which consists of endorser nodes. These endorser nodes are important for the verification and validation of the posted product reviews. We also provide a security analysis of EthReview by evaluating its resilience and robustness against the adversary model. For real-time testing and validation, our solution is implemented on *Ganache* and *Ropsten TestNet* along with *Metamask* for performing transactions. Moreover, a user-friendly web-interface is also integrated for interacting with the smart contracts. We estimated the operational cost of our system in terms of gas costs utilized to perform major transactions. Since we don't have any state of the art product review system developed on Ethereum to compare our operational cost with. We believe that the gas cost incurred to perform transactions in EthReview is reasonable for buyers, sellers, and endorsers without causing any additional major cost overhead. While this paper provides some foundational ideas for the development of a decentralized Ethereum-based product review system, there is still a lot of scope for further research. Especially the bootstrapping of the endorsement can be enhanced. As future work, we are in the process of evaluating and enhancing the scalability of our solution through implementation on private blockchain networks. Moreover, variable model for *ReviewFund* and other strategies will be incorporated in future work to make our proposed solution economically viable.

8. ACKNOWLEDGMENTS

This research project was partially funded by Blockchain Research Lab at Information Technology University (ITU), Lahore, Pakistan.

References

[1] P. Chatterjee, Online reviews: do consumers use them?, Proceedings of Association for Consumer Research (ACR) (2001).

[2] Statista.com, For each of the following circumstances, how important is it to read online reviews before purchasing a product or selecting a service provider?, 2017. URL: https://www.statista.com/statistics/713258/online-review-importance-circumstances-usa/.

[3] Statista.com, Impact on user-generated content such as customer reviews and ratings according to online shoppers in the united sates as of march 2017, 2017. URL: https://tinyurl.com/sm2pdbh.

[4] C. Dellarocas, The digitization of word of mouth: Promise and challenges of online feedback mechanisms, Manage. Sci. 49 (2003) 1407–1424. URL: http://dx.doi.org/10.1287/mnsc.49.10.1407.17308. doi:10.1287/mnsc.49.10.1407.17308.

[5] Pivotrev.com, Pivotrev, 2018. URL: https://pivotrev.com/.

[6] T. T. Mareike MÃűhlmann, A. Graul, Leveraging trust on sharing economy platforms: reputation systems, blockchain technology and cryptocurrencies, Edward Elgar Publishing, 2019. URL: "https://doi.org/10.4337/9781788110549.00033".

[7] S. Tadelis, Reputation and feedback systems in online platform markets, Annual Review of Economics 8 (2016) 321–340. URL: https://doi.org/10.1146/annurev-economics-080315-015325. doi:10.1146/annurev-economics-080315-015325. arXiv:https://tinyurl.com/y2k3prlj.

[8] A. Fradkin, E. Grewal, D. Holtz, M. Pearson, Bias and reciprocity in online reviews: Evidence from field experiments on airbnb, in: Proceedings of the Sixteenth ACM Conference on Economics and Computation, EC '15, ACM, New York, NY, USA, 2015, pp. 641–641. URL: http://doi.acm.org/10.1145/2764468.2764528. doi:10.1145/2764468.2764528.

[9] M. Voss, Privacy preserving online reputation systems, in: Y. Deswarte, F. Cuppens, S. Jajodia, L. Wang (Eds.), Information Security Management, Education and Privacy, Springer US, Boston, MA, 2004, pp. 249–264.

[10] L. Liu, M. Munro, Systematic analysis of centralized online reputation systems, Decision Support Systems 52 (2012) 438 – 449. URL: http://www.sciencedirect.com/science/article/pii/S0167923611001667. doi:https://doi.org/10.1016/j.dss.2011.10.003.

[11] C. Lam, Applying blockchain technology to online reviews, LSE Business Review Blog (2017).

[12] K. Salah, A. Alfalasi, M. Alfalasi, A blockchain-based system for online consumer reviews, IEEE INFOCOM International Workshop on Cryptocurrencies and Blockchains for Distributed Systems (2019).

[13] D. Martens, W. Maalej, Reviewchain: Untampered product reviews on the blockchain, in: 2018 IEEE/ACM 1st International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB), 2018, pp. 40–43.

[14] T. Fornaciari, M. Poesio, Identifying fake Amazon reviews as learning from crowds, in: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, 2014, pp. 279–287.

[15] N. Jindal, B. Liu, Opinion spam and analysis, in: Proceedings of the 2008 International Conference on Web Search and Data Mining, WSDM '08, ACM, New York, NY, USA, 2008, pp. 219–230. URL: http://doi.acm.org/10.1145/1341531.1341560. doi:10.1145/1341531.1341560.

[16] A. Mukherjee, V. Venkataraman, B. Liu, N. Glance, What yelp fake review filter might be doing?, in: Seventh international AAAI conference on weblogs and social media, 2013.

[17] N. Hu, L. Liu, V. Sambamurthy, Fraud detection in online consumer reviews, Decision Support Systems 50 (2011) 614 – 626. URL: http://www.sciencedirect.com/science/article/pii/S0167923610001363. doi:https://doi.org/10.1016/j.dss.2010.08.012, on quantitative methods for detection of financial fraud.

[18] Y. Cai, D. Zhu, Fraud detections for online businesses: a perspective from blockchain technology, Financial Innovation 2 (2016) 20. doi:10.1186/s40854-016-0039-4.

[19] C. Dellarocas, Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior, in: Proceedings of the 2nd ACM Conference on Electronic Commerce, EC '00, Association for Computing Machinery, New York, NY, USA, 2000, p. 150âĂŞ157. URL: https://doi.org/10.1145/352871.352889. doi:10.1145/352871.352889.

[20] P. Viennas, Ethereum consensus and scalability (blockchain series - part iii), 2018. URL: https://tinyurl.com/y5owcsxg.

[21] K. Hoffman, D. Zage, C. Nita-Rotaru, A survey of attack and defense techniques for reputation systems, ACM Comput. Surv. 42 (2009). URL: https://doi.org/10.1145/1592451.1592452. doi:10.1145/1592451.1592452.

[22] F. ul Hassan, A. Ali, S. Latif, J. Qadir, S. Kanhere, J. Singh, J. Crowcroft, Blockchain and the future of the internet:a comprehensive review, 2019. arXiv:1904.00733.

[23] G. Wood, Ethereum: A secure decentralised generalised transaction ledger, Ethereum project yellow paper 151 (2014) 1–32.

[24] N. Szabo, The idea of smart contracts, 1997, URL http://szabo. best. vwh. net/smart_contracts_idea. html (1997).

[25] C. Dannen, Introducing Ethereum and solidity, volume 1, Springer, 2017.

[26] K. Salah, M. H. U. Rehman, N. Nizamuddin, A. Al-Fuqaha, Blockchain for ai: Review and open research challenges, IEEE Access 7 (2019) 10127–10149. doi:10.1109/ACCESS.2018.2890507.

[27] A. Rosic, What is ethereum gas? [the most comprehensive step-by-step guide ever!], 2018. URL: https://blockgeeks.com/guides/ethereum-gas/.

[28] J.-S. Park, T.-Y. Youn, H.-B. Kim, K.-H. Rhee, S.-U. Shin, Smart contract-based review system for an iot data marketplace, Sensors 18 (2018) 3577. URL: http://dx.doi.org/10.3390/s18103577. doi:10.3390/s18103577.

[29] EIPS-Ethereum, Ethereum improvement proposals, 2019. URL: https://eips.ethereum.org/erc.

[30] S. Tadelis, The economics of reputation and feedback systems in e-commerce marketplaces, IEEE Internet Computing 20 (2016) 12–19.

[31] R. Ramachandiran, Using blockchain technology to improve trust in ecommerce, doi:10.13140/RG.2.2.29324.00646 (2018) 14.

[32] R. Dennis, G. Owenson, Rep on the roll: a peer to peer reputation system based on a rolling blockchain, International Journal for Digital Society 7 (2016) 1123–1134. doi:10.20533/ijds.2040.2570.2016.0137, authors retain publishing rights without restrictions: http://infonomics-society.org/ijds/.

[33] R. Dennis, G. Owenson, Rep on the block: a next generation reputation system based on the blockchain, in: Proceedings of the 10th International Conference for Internet Technology and Secured Transactions (ICITST), IEEE, 2016, pp. 131–138. doi:10.1109/ICITST.2015.7412073, 10th International Conference for Internet Technology and Secured Transactions, ICITST 2015 ; Conference date: 14-12-2015 Through 16-12-2015.

[34] Privbox: Verifiable decentralized reputation system for online marketplaces, Future Generation Computer Systems 89 (2018) 44 – 57. URL: http://www.sciencedirect.com/science/article/pii/S0167739X17330315. doi:https://doi.org/10.1016/j.future.2018.05.069.

[35] N. Busom, R. Petrlic, F. SebÃĬ, C. Sorge, M. Valls, A privacy-preserving reputation system with user rewards, Journal of Network and Computer Applications 80 (2017) 58 – 66. URL: http://www.sciencedirect.com/science/article/pii/S1084804516303320. doi:https://doi.org/10.1016/j.jnca.2016.12.023.

[36] D. Carboni, Feedback based reputation on top of the Bitcoin blockchain, CoRR abs/1502.01504 (2015). URL: http://arxiv.org/abs/1502.01504. arXiv:1502.01504.

[37] A. Schaub, R. Bazin, O. Hasan, L. Brunie, A trustless privacy-preserving reputation system, in: J.-H. Hoepman, S. Katzenbeisser (Eds.), ICT Systems Security and Privacy Protection, Springer International Publishing, Cham, 2016, pp. 398–411. doi:https://doi.org/10.1007/978-3-319-33630-5_27.

[38] A. Aravazhi irissappane, S. Jiang, J. Zhang, Towards a comprehensive testbed to evaluate the robustness of reputation systems against unfair rating attacks, volume 872, 2012.

[39] A. Hertig, Bitcoin Cash Miners Undo AttackerâĂŹs Transactions With âĂŸ51% AttackâĂŹ, CoinDesk, 2019. URL: https://tinyurl.com/y27kqc42.

[40] M. Nesbitt, Deep Chain Reorganization Detected on Ethereum Classic (ETC), The Coinbase Blog, 2019. URL: https://tinyurl.com/y7b32t49.

[41] M. Nesbitt, Vertcoin (VTC) was successfully 51% attacked, Medium, 2018. URL: https://medium.com/coinmonks/vertcoin-vtc-is-currently-being-51-attacked-53ab633c08a4.

[42] D. Kuhn, Blockchain Bites: Ethereum Classic Attacked, Electrum Wallet Drained and Taxable Microtasks, The Coinbase Blog, 2020. URL: https://www.coindesk.com/blockchain-bites-ethereum-classic-electrum-tax.

[43] Ghash.io, BitcoinWiki, ???? URL: https://en.bitcoinwiki.org/wiki/GHash.IO#51.25_attack_controversy.

[44] S. Sayeed, H. Marco-Gisbert, Assessing blockchain consensus and security mechanisms against the 51% attack, Applied Sciences 9 (2019) 1788. URL: http://dx.doi.org/10.3390/app9091788. doi:10.3390/app9091788.

[45] S. Kim, B. Kim, H. J. Kim, Intrusion detection and mitigation system using blockchain analysis for bitcoin exchange, in: Proceedings of the 2018 International Conference on Cloud Computing and Internet of Things, CCIOT 2018, Association for Computing Machinery, New York, NY, USA, 2018, p. 40âĂŞ44. URL: https://doi.org/10.1145/3291064.3291075. doi:10.1145/3291064.3291075.

[46] H. Chen, M. Pendleton, L. Njilla, S. Xu, A survey on ethereum systems security: Vulnerabilities, attacks, and defenses, ACM Computing Surveys (CSUR) 53 (2020) 1–43.

[47] S. Tikhomirov, E. Voskresenskaya, I. Ivanitskiy, R. Takhaviev, E. Marchenko, Y. Alexandrov, Smartcheck: Static analysis of ethereum smart contracts, in: 2018 IEEE/ACM 1st International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB), 2018, pp. 9–16.

[48] E. Wiki, Ethereum sharding, 2019. URL: https://eth.wiki/sharding/Sharding-FAQs.

[49] S. Kim, Y. Kwon, S. Cho, A survey of scalability solutions on blockchain, in: 2018 International Conference on Information and Communication Technology Convergence (ICTC), IEEE, 2018, pp. 1204–1207.

[50] C. Saraf, S. Sabadra, Blockchain platforms: A compendium, in: 2018 IEEE International Conference on Innovative Research and Development (ICIRD), IEEE, 2018, pp. 1–6.

AUTHOR BIOGRAPHY

Maryam Zulfiqar is working as Research Assistant in the Blockchain Research Lab at Information Technology University (ITU) in Lahore, Pakistan. Also, worked as Instructor in the University of Management & Technology Lahore Pakistan. Current research interests are blockchain based projects, security and networks. She has earned a B.S in Computer Science from the International Islamic University of Islamabad, Pakistan and received gold medal for her academic performance. She is currently pursuing a M.S. in Computer Science at the Information Technology University in Lahore, Pakistan.

Filza Tariq has completed her bachelors in computer science in 2018 from Lahore College for Women University and currently pursuing her masters in computer science at Information Technology University. She is also working as Research Assistant in the Blockchain Research Lab at Information Technology University (ITU) in Lahore, Pakistan. She has led various community initiatives for the inclusion of blockchain technology that includes delivering community sessions on blockchain development and co-founding a nationwide blockchain community. Filza possesses expertise in distributed application development on various blockchain platforms including Ethereum and Azure blockchain. Her research interest includes blockchain and information security.

Dr. Muhammad Umar Janjua is an Assistant Professor in the Department of Computer Science at the Information Technology University (ITU) in Lahore, Pakistan. He is also the Director of Blockchain Research Lab at ITU. He worked at Microsoft Corporation as a software engineer from 2008 to 2017 in the Windows Security R&D group. He has been the primary developer in preparing several mitigation packages for the entire Windows ecosystem. He completed his Ph.D. from Cambridge University. His research interests include static analysis, program verification and synthesis, big data security, and applied cryptography.

Dr. Adnan Noor Mian is an Associate Professor at Information Technology University (ITU), Lahore, Pakistan where he leads the Internet of Things (IoT) research lab. He received his PhD in Computer Engineering in 2009 from Sapienza University of Rome, Italy in the field of distributed systems. In 2018-19 he has been a postdoc researcher in the Department of Computer Science and Technology, University of Cambridge, UK. His research interests include wireless sensor and ad hoc networks, Internet of Things (IoT), mesh networks, V2X communication, distributed algorithms, mobile and distributed systems, cloud and fog computing, in which he has more than 40 publications in the leading venues of the field. He is also serving in a number of technical program committees of international conferences and journals. With more than 20 years of teaching experience, Dr. Adnan has taught a variety of courses at the graduate and undergraduate level. He is the founding chairperson of Department of Computer Science in the Information Technology University, Lahore and has taken a number of initiatives to promote research culture in the University. In 2016 he was awarded the Frogh-e-Taleem Award from Idara Frogh-e-Taleem by the government of Punjab, Pakistan and in 2017 he was awarded by the Chief Minister of Punjab, Pakistan. Recently he is selected as Senior Associate of International Centre of Theoretical Physics (ICTP), Trieste, Italy in the field of IoT. He is the first one, not only from Pakistan but also from other counties, to get this position in the field of IoT.

Dr. Adnan Qayyum received the bachelorâĂŹs degree in Electrical (Computer) Engineering from the COMSATS Institute of Information Technology Wah, Pakistan, in 2014, and the M.S. degree in Computer Engineering (Signal and Image Processing) from the University of Engineering and Technology, Taxila, Pakistan, in 2016. He is currently pursuing the Ph.D. degree in Computer Science with Information Technology University, Lahore Pakistan. His research interests include autonomous vehicles, healthcare, deep/machine learning (ML), and security of ML.

Dr. Junaid Qadir is the director of the IHSAN-ICTD; Human Development; Systems; Big Data Analytics; Networks-Research Lab and the Chairperson of the Electrical Engineering Department at the Information Technology University (ITU) of Punjab in Lahore, Pakistan. His primary research interests are in the areas of computer systems and networking, applied machine learning, using ICT for development (ICT4D); and engineering education. He has published more than 100 peer-reviewed articles at various high-quality research venues including more than 50 impact-factor journal publications at top international research journals including IEEE Communication Magazine, IEEE Journal on Selected Areas in Communication (JSAC), IEEE Communications Surveys and Tutorials (CST), and IEEE Transactions on Mobile Computing (TMC). He was awarded the highest national teaching award in Pakistan-the higher education commissionâĂŹs (HEC) best university teacher award-for the year 2012-2013. He has been appointed as ACM Distinguished Speaker for a three-year term starting from 2020. He is a senior member of IEEE and ACM.

Dr. Muhammd Hassan completed his PhD in cybersecurity from the Brain, Mind and Computer Science (BMCS) department at the University of Padua, Italy in 2019. Currently, he is working at the National Center of Cyber Security (NCCS) as a Team Lead research at Blockchain Lab Information Technology University, Pakistan. He is also a member of the Security and PRIvacy Through Zeal (SPRITZ) research group, Italy. At a high level, his research focuses on security, privacy and access control issues in Future Internet Architectures (FIA), specifically, Information-Centric Networking (NDN/CCN) and related studies, such as secure integration of existing technologies (e.g., blockchain, future mobile networks, multimedia streaming) in FIA. Besides, he has completed bachelors in Electrical (Telecommunication) Engineering from COMSATS Institute of Information Technology, Pakistan and masters in Computer Network Engineering from HALMSTAD university, Sweden.

Dr. Falak Sher earned his PhD in Software Modelling & Verification from RWTH Aachen University, Germany. He did his post-doctoral research at Fortiss GmbH – national research institute of Germany. His areas of expertise include analysis of reactive, stochastic, real-time, and hybrid systems, reduction techniques for probabilistic systems, game theory, formal software verification, in particular model checking, formal modeling and analysis of distributed computing, mathematical logic, automata theory. On application side he worked in Blockchain, Machine Learning particularly reinforcement learning. He has published in multiple international conferences related to formal methods, blockchain and AI/ML.

Declaration of interests

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Credit Author Statement

Maryam Zulfiqar:
Conceptuliazation, Methodology, Software, Writing âĂŞ Original Draft, Writing âĂŞ Review & Editing
Filza Tariq:
Conceptuliazation, Methodology, Software, Writing âĂŞ Original Draft, Writing âĂŞ Review & Editing
Dr. Muhammad Umar Janjua:
Supervision, Methodology, Project Administration, Writing- Review & Editing
Dr. Adnan Noor Mian:
Writing- Review & Editing
Dr. Adnan Qayyum:
Writing- Review & Editing
Dr. Junaid Qadir:
Writing- Review & Editing
Dr. Muhammad Hassan:
Writing- Review & Editing
Dr. Falak Sher: Writing- Review & Editing