Anonymous and Verifiable Reputation System for E-commerce Platforms based on Blockchain

Li, Meng; Zhu, Liehuang ; Zhang, Zijian; Lal, Chhagan; Conti, Mauro; Alazab, Mamoun

DOI
10.1109/TNSM.2021.3098439

Publication date
2021

Document Version
Accepted author manuscript

Published in
IEEE Transactions on Network and Service Management

Citation (APA)
Li, M., Zhu, L., Zhang, Z., Lal, C., Conti, M., & Alazab, M. (2021). Anonymous and Verifiable Reputation System for E-commerce Platforms based on Blockchain. *IEEE Transactions on Network and Service Management*, *18*(4), 4434-4449. Article 9490665. https://doi.org/10.1109/TNSM.2021.3098439

Anonymous and Verifiable Reputation System for E-commerce Platforms based on Blockchain

Meng Li, *Member, IEEE*, Liehuang Zhu*, *Member, IEEE*, Zijian Zhang, *Member, IEEE*, Chhagan Lal, *Member, IEEE*, Mauro Conti, *Senior Member, IEEE*, Mamoun Alazab, *Senior Member, IEEE*

***Abstract*—E-commerce platforms incorporate reputation systems that allow customers to rate suppliers following financial transactions. Existing reputation systems cannot defend the centralized server against arbitrarily tampering with the supplier's reputation. Furthermore, they do not offer reputation access across platforms. Rates are faced with privacy leakages because rating activities are correlated with privacy (e.g., identity and rating). Meanwhile, raters could be malicious and initiate multiple rating attacks and abnormal rating attacks. Determining how to address these issues have both research and practical value.**

In this paper, we propose a blockchain-based privacy-preserving reputation system for e-commerce platforms named RepChain; our system allows cross-platform reputation access and anonymous and private ratings. Using RepChain, all e-commerce platforms collaborate and share users' reputations by co-constructing a consortium blockchain and modeling the rating process as a finite state machine. In particular, we facilitate one-show anonymous credentials constructed from two-move blind signatures to protect customers' identities and resist multiple rating attacks, leverage zero-knowledge range proof to verify the correctness of ratings and defend against abnormal rating attacks, design a secure sum computation protocol among nodes to update reputations, and verify ratings via batch processing and consensus hashes. Finally, we demonstrate the security and privacy of RepChain via a formal analysis and evaluate its performance based on Ethereum test network.

***Index Terms*—E-commerce platforms, rating, privacy, security, blockchain.**

I. Introduction

E-commerce platforms, such as eBay, AirbnB, Yelp, and Stack Overflow, have become increasingly popular and are considered forerunners of our future online business and sharing economy. For example, eBay, which is one of the biggest sharing economy company, processes over one billion transactions everyday [1] and is now worth more than $2 billion [2]. As e-commerce platforms permeate our daily lives, users will be able to conduct business with other users worldwide and achieve peer-to-peer exchange. However, implementing such a platform cannot work without a powerful reputation system.

A reputation system collects information about a user's feedbacks on financial transactions based on perceived utility. The feedbacks increase or decrease user reputation which contributes to a better understanding between financial entities before new transactions take place. Here "user" encompasses both the customer and the supplier. Because customers prefer to obtain services from highly regarded suppliers, building a robust and fair reputation system will improve the buying confidence of customers, drive the sales growth of suppliers, reduce transaction risks, and ultimately improve the overall quality of online markets [3]. Given these considerations, commercial companies have an interest in implementing a robust reputation system. One typical reputation system in the e-commerce domain is used by eBay [4].

Even though existing centralized reputation systems [5], [6], [7] provide some benefits, they suffer from the following problems. **Centralization**: Users' reputations are stored and updated on a centralized server, which creates a single point of failure/attack. In addition, such a model is prone to be falsified by a malicious platform without the users' consent. **Isolation**: When a customer requests a service from a supplier of another platform, the supplier cannot access the reputation of this customer on time. This is because the customer's reputation is encapsulated in an isolated database that other platforms cannot access. Additionally, there are many credit scoring models [8], which can hinder cross-platform access. **Lack of information fuse**: Stemming from the two problems above, it can be extremely challenging to predict behaviors and evaluate the trustworthiness of users to reduce financial risks and damage [9]. As an example, in May 2018, a male Didi driver, who had a record of several sexual harassment complaints, killed a female passenger [10]. Such an unfortunate incident might have been prevented if it had been possible to require the reputation (from all possible sources) of business partners in advance, given that existing reputation systems do not support predictability.

All these challenges make it highly desirable to develop a transparent, tamper-resistant, and cross-platform reputation system. To address these challenges, one possible solution could be blockchain [11], [12]. Originally, blockchain is a fundamental technology beneath Bitcoin. But nowadays it has been adopted in different domains including Artificial Intelligence [13], Internet of Things [13], 5G [14], digital

Meng Li is with Key Laboratory of Knowledge Engineering with Big Data (Hefei University of Technology), Ministry of Education; School of Computer Science and Information Engineering, Hefei University of Technology; Anhui Province Key Laboratory of Industry Safety and Emergency Technology; and Intelligent Interconnected Systems Laboratory of Anhui Province (Hefei University of Technology). (Email: mengli@hfut.edu.cn)

Liehuang Zhu (Corresponding Author) and Zijian Zhang are with the School of Computer Science and Technology at the Beijing Institute of Technology. Zijian Zhang is also with the School of Computer Science at the University of Auckland, New Zealand. (Email: {liehuangz, zhangzijian}@bit.edu.cn)

Chhagan Lal is with Department of Intelligent Systems, CyberSecurity Group, TU Delft, Netherlands. (Email: c.lal@tudelft.nl)

Mauro Conti is with the Department of Mathematics, University of Padua, 35131 Padua, Italy. (e-mail: conti@math.unipd.it)

Mamoun Alazab is with College of Engineering, IT and Environment, Charles Darwin University, Australia. (Email: alazab.m@ieee.org)

twins [15], and supply chain [16]. By introducing blockchain into the reputation system, we could achieve a public and tamper-resistant record of ratings and reputations, as well as the access to the reputations of suppliers from different platforms. Due to the fact that (1) public blockchains face information leakage [17], and (2) the single-manager mode of private blockchains (which may arbitrarily tamper with the blockchain) is not compatible with the multiple platforms setting; here, we propose to use a consortium blockchain (CBC) [18], [19]. Specifically, different platforms agree to collaborate and co-establish a CBC for the reputation system. Such a CBC verifies all rating transactions sent by customers. Rating transactions are packed into data blocks using an elected node according to the group consensus. The data blocks are chained in a cryptographically indisputable way such that no one can tamper with the rating transactions. A CBC can break the data barrier across different platforms and build a harmonious online shopping ecosystem. Such a system will benefit platforms and users with respect to references, management, and motivation [20].

Despite the benefits of blockchain, a direct implementation of such a technology recording all ratings on the public ledger will result in a lack of *rating privacy* [6], [7] and would enable some security attacks. First, a rating is attached to the identity of the customer, which is considered sensitive. Second, a rating is private because how a customer rates a product reveals her/his preferences such that a low rating may even incur retaliation from a spiteful supplier [9]. Third, a set of the same customer's ratings is closely related to her/his commercial activities or even financial status [21]. Unlinkability must be guaranteed such that two rating transactions from the same rater cannot be linked. Next, some malicious customers could initiate a multiple rating attack, i.e., rate a supplier multiple times after one financial transaction, and an abnormal rating attack, i.e., submits a rating outside of the normal range. As a consequence, the system fairness and rating correctness are undermined. Finally, it is not easy to securely compute the average reputation in a distributed blockchain network if each rating is encrypted. Therefore, the significant **technical challenges** of designing a blockchain-based reputation platform are enabling the collaboration of different platforms while preserving privacy, unlinkability, and resisting security attacks in an untrusted and distributed network.

To address these challenges, we propose RepChain: a blockchain-based privacy-preserving reputation system for e-commerce platforms. Using RepChain, platforms will collaborate to share suppliers' reputations and the rating process is modeled as a finite-state machine in smart contracts. The blockchain nodes (hereinafter referred to as nodes), which execute the consensus mechanism to maintain the distributed ledger, assist in updating the raters's reputation by using secure multiparty computation. In addition, the system should serve as a disincentive to users to engage in misconduct and encouraging positive user behaviors. *To the best of our knowledge,* RepChain *is the first system to offer both reputation access and rating privacy across multiple platforms in a decentralized environment.* An overview of RepChain is shown in Fig. 1 in which several e-commerce platforms, such as Amazon and Best Buy, construct a consortium blockchain to record all rating-related transactions in public. The key contributions are as follows.

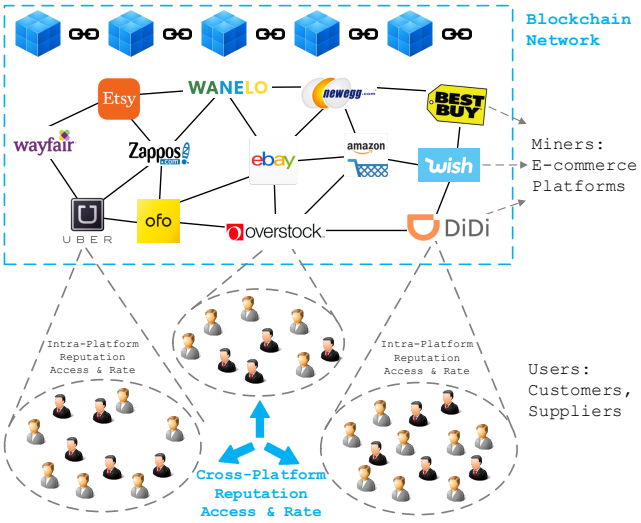


Fig. 1. Overview of RepChain

- We propose a novel decentralized framework for an e-commerce platform reputation system based on a consortium blockchain. Under this framework, we focus on the rating related activities and define the rating transaction. The proposed framework enhances the service availability by allowing access to reputation scores across platforms and calculating the overall reputation from the ratings. The system is also privacy-preserving and resistant to multiple rating and abnormal rating attacks.
- We design a concrete scheme to guarantee security and privacy protection. Specifically, we facilitate blind signatures [22] to protect the rater's identities and resist multiple rating attacks. We leverage zero-knowledge range proof [23] to defend against abnormal rating attacks. Next, we establish a secure sum computation protocol [24], [25], and threshold Paillier cryptosystem [26] to hide ratings and obtain the sum of ratings. To improve the verification efficiency, we adopt consensus hashing [27] to verify the rating transactions.
- We formally prove the privacy and security of the proposed scheme. We evaluate the performance of the proposed scheme by implementing a prototype based on Ethereum [28] test network to demonstrate its feasibility and efficiency. We also compare its computational costs and communication overhead with existing work.

The rest of this paper is organized as follows. We review related work in Section II. We define the system model, security model, design goals, and technical challenges in Section III. Section IV briefly revisits the preliminaries. We present the proposed RepChain system in Section V, followed by security and privacy analysis in Section VI and performance evaluation in Section VII. Lastly, we discuss some issues in Section X and conclude our work in Section IX.

II. Related Work

Hasan et al. [3] proposed a distributed reputation protocol (DPPR for short) where a target user (ratee) interacted with source users (raters) who had assigned it private feedbacks. The protocol enables a user to query the reputation of other users as the mean of the private feedbacks while not disclosing feedbacks of source users. However, this scheme suffers from a high computational cost: each source user has to encrypt a share of his private feedback twice, i.e., first to encrypt it with a recipient's public key and then to encrypt it with his public key. Additionally, each source user has to prove that the two ciphertexts contain the same plaintext through plaintext-equality zero-knowledge proof.

Tassos et al. [29] presented a reputation protocol StR atop the Paillier cryptosystem [30]. It enables participants to submit their ratings securely. A querying user creates a set of k participants. Each participant splits a random number into k pieces and sends an encrypted piece to other nodes, and computes a blinded vote based on decryptions of received ciphertexts. Then, each participant encrypts the blinded vote with the public key of the querying user, computes a sum of previous ratings, and forwards it to the next participant until it reaches the querying user. The querying user can decrypt the result and obtains the sum of all ratings.

Blömer et al. [5] defined models for anonymous and secure reputation systems and proposed such a system (ASRS for short) based on BBS group signatures and verifier-local revocation group signatures. Anonymity and authentication are achieved. However, each rating is public and the provided public linkability has enabled anyone can decide whether two ratings for the same product are submitted by the same customer.

Schaub et al. [31] presented a trustless privacy-preserving reputation system (TPPR for short) based on a blockchain for e-commerce applications. Before a customer initiates a transaction, he first asks the service provider with sufficient balance to blindly sign a token [32]. Then the customer unblinds the token to broadcast a message including the token and a rating. But they only consider customers' ratings towards the service provider, and the rating of a customer is revealed during a transaction.

Dennis et al. [33] presented a generalized reputation system based on a blockchain for multiple networks. But they only used a single-dimensional reputation on the blockchain with 1/0 for a positive/negative review. When the nodes check the validity of each transaction, they have to request a signed proof from each user involved in the transaction. However, it requires users to be online.

Zhai et al. [6] proposed an anonymous reputation system AnonRep to provide identity anonymity, unlinkability, and private rating by using verifiable shuffles, linkable ring signatures, and homomorphic crypto. AnonRep users submit their ratings anonymously while guaranteeing security against duplicate feedbacks or score tampering. No entity could link ratings to any user identity. However, the rating is stored in plaintext.

Azad et al. [7] proposed a privacy-preserving reputation system PrivBox to securely and privately compute the suppliers' reputation by leveraging homomorphic cryptosystem and noninteractive zero-knowledge proof. PrixBox can protect customer privacy, check whether the customer rating is in a prescribed range, and ensures that the computed statistics are verifiable. However, it still builds on a centralized model and only two ratings (positive and negative) are supported.

DREP [20] is a blockchain-based system based which quantifies reputation for trading, investment and data sharing among different E-commerce platforms. It is a powerful reputation system built atop a reputation quantification, reputation monetization, and voting. Although it allows users to choose whether their reputation values can be seen by the platform and other users, which could be a step back for online transactions, but it does not protect the ratings.

Casino et al. [34] leveraged blockchain, decentralized locality sensitive hashing classification, and recommendation methods to support a decentralized recommender system (RS) with several features while preserving user's privacy. It allows a user to collect data and execute a bucketization procedure with result being stored in a blockchain and InterPlanetary File System (IPFS). Next, the data wrapper collects data pinned by users, and users compute recommendations. In comparison, the proposed scheme is a good recommendation system while our work is a rating system. We compare existing work in terms of decentralization, privacy, and security in Table I.

III. Problem Statement

In this section, we define the system model in Section III.A, security model in Section III.B, and design goals in Section III.C.

A. System Model

Our RepChain system is a reputation system auxiliary to the original e-commerce systems and the proposed model consists of four entities: n_1 customers, n_2 suppliers, n_3 platforms and one certificate authority. The system model of RepChain is displayed in Fig. 2 and some key notations are explained in Table II.

Customer (C) receives a product or service sold by a supplier through a financial transaction or exchange for some valuable assets. After the financial transaction, the customer prepares and submits a rating of a supplier to the CBC in the form of a rating transaction. Ratings are real numbers ranging from 0 to 10.

Supplier (S) sells a product or service to a customer through a similar transaction mentioned above. After the financial transaction, the supplier awaits a rating transaction from the customer.

Platform (P) assists them in exchanging products, services, and money. Platforms are nodes in the CBC. Platforms receive and verify rating transactions. A new platform can join the reputation system after being acknowledged by existing platforms and registered by CA. A platform can be removed from the reputation system by CA's broadcasting. We note that both customer and supplier could register to more than one platform with their metadata, i.e., email addresses or cellphone numbers. In cross-platform collaboration, we assume that

TABLE I
BRIEF COMPARISON OF REPCHAIN AND EXISTING WORK

Property	DPPR [3]	StR [29]	ASRS [5]	TPPR [31]	PrivBox [7]
Decentralized	√	√	×	√	√
Transparency	√	√	×	√	√
Rating privacy	√	√	×	×	√
Identity Privacy	√	√	√	√	√
Unlinkability	√	√	×	√	√
Resistance to multiple rating attack	×	×	√	√	×
Resistance to abnormal rating attack	×	×	×	×	√

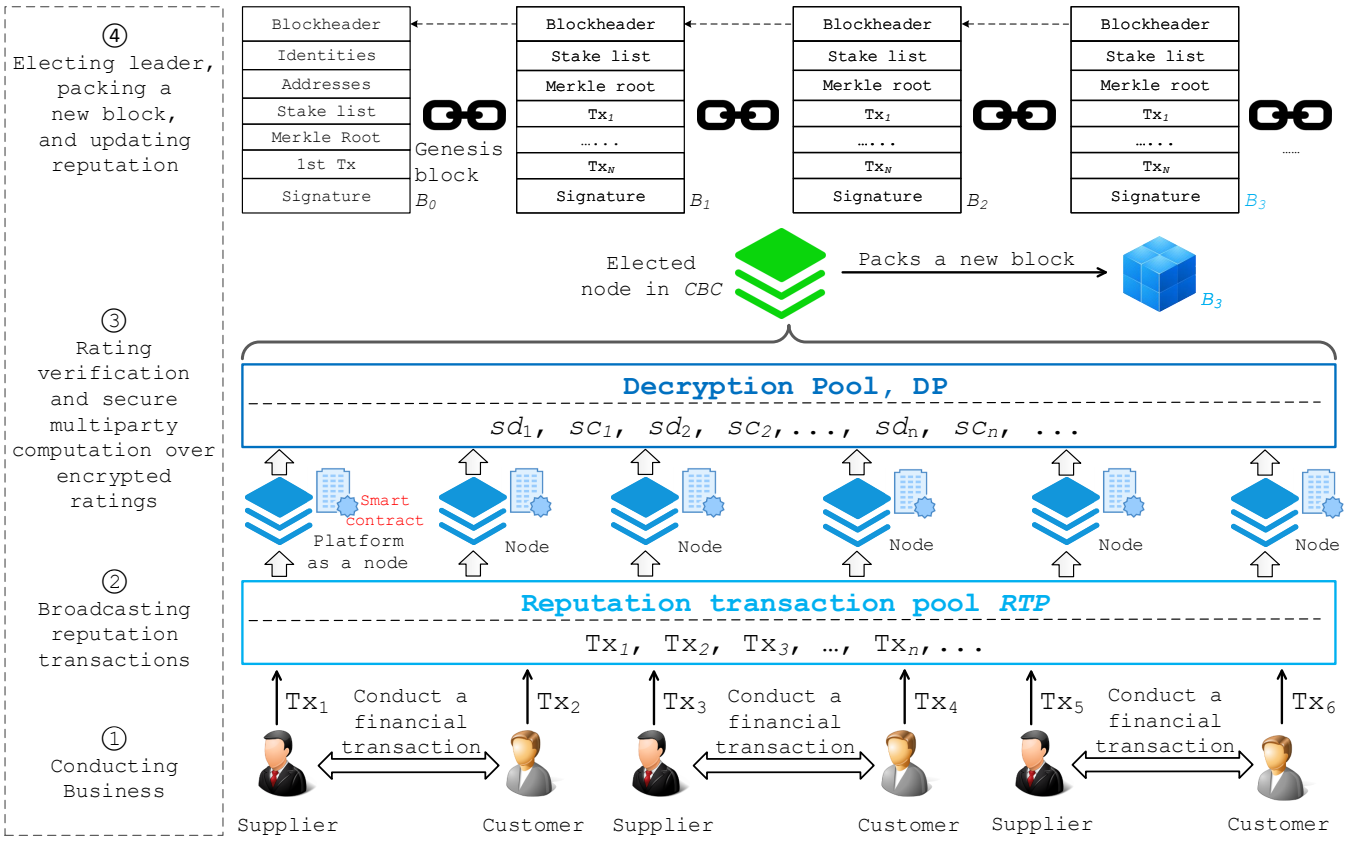


Fig. 2. The system model of RepChain

different platforms integrate their ratees' accounts through the same metadata. If the ratee has used different metadata in registration, the platforms will ask her/him to link the account in advance.

Certificate Authority (CA) is an e-commerce business association that is co-founded by all the platforms. CA is responsible for generating system parameters and cryptographic keys for users and platforms. It receives registration requests from users. It does not conflict with the decentralized feature of blockchain because it stays offline after system initialization and entity registration.

Note: *Rater* refers to the one who can submit a rating and *ratee* refers to the one who receives a rating. A customer and a supplier can be either a rater or ratee in a rating transaction. **For a concise description, we will use the customer as the rater and use the supplier as the ratee.**

Some core notions are explained as follows. **Reputation** is a real number indicating how a supplier behaves during past financial transactions. It should be between a range $[LB, UB]$ where LB and UB represent the lower and upper bound of rating, respectively. **Rating** is a real number produced by a customer towards a supplier after they engage in a financial transaction. Each rating should stay in the same range as its reputation. We note that RepChain also supports textual comments. **Rating transaction** is used to record a rating from a customer to a supplier. It has one input and one output. The supplier's reputation is traceable through a transaction chain. **Block** is a package of rating transactions and blocks are chained together one by one. Each new block is created by a winning node, i.e., an elected node based on group consensus PoS [35]. **Node** is a platform performing secure multiparty computation via smart contracts. Nodes compete with other nodes to be the elected node who creates a new block in the current period. **Smart contract** is a segment of codes automatically executed by nodes [36]. It securely participates in computing the average reputation of suppliers

without revealing ratings.

The main operations are explained as follows.

- Access. Prior to initiating a financial transaction, a customer looks up a supplier's reputation via inputting the supplier's public key to a CBC interface and receives a value of reputation.
- Rate. After the financial transaction is complete, the customer obtains a blind-signed signature and unblinds it to obtain a rating credential. Then the customer generates a rating, i.e., real number in a range $[LB, UB]$. Next, the customer encrypts the rating under the system public key to obtain a ciphertext. Finally, the customer broadcasts a rating transaction to a rating transaction pool RTP.
- Mine. Each node calculates a share decryption of ratings and a share combining, i.e., product, of the decryptions and sends it to a decryption pool DP. In this process, the nodes can calculate multiple decryptions for many transactions expecting to be the elected node. Only the nodes are required to download the entire CBC.
- Update. For each supplier, the elected node will choose and verify decryptions in DP and send the combined sum result to corresponding suppliers. Finally, the supplier updates his received number of rating nr, reputation rep, and liveness degree ld.

TABLE II
KEY NOTATIONS

Notation	Definition
C, S, CA	Customer, supplier, certificate authority
$P, \hat{P}$	E-commerce platform, Elected platform
RTP, DP	Rating transaction pool, Decryption pool
$\kappa; \hat{p}, p, q, p', q'$	Security parameter; Prime number
$\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e$	Multiplicative cyclic group, bilinear paring
g_1, g_2	Group generator
H, f, sp	Hash function, polynomial, plaintext limit
n_1, n_2, n_3	Number of customers/suppliers/platforms
rep, rt	Reputation value, rating
Φ, LB, UB	Rating range, lower bound, upper bound
ℓ, ℓ'	Key vector length, rating vector length
$\mathsf{sk}; \mathsf{pk}_{\mathcal{R}}, \mathsf{pk}$	Secret key; public key
$Com, \overline{RT}$	Commitment, blinded commitment
σ, ts, Tx	Signature, timestamp, rating transaction
$\mathsf{F}, \hat{P}$	Leader election function, elected platform
n_{Tx}	Number of transactions in a block
pk	Public keys of supplier (ratee)
ss, vk	Secret share, verification key
sd, sc	Share decryption, share combining
N	Number of share decryptions
$CH, \mathcal{P}$	Consensus hash, a series of prior CHs

B. Security Model

Security threats come from internal and external adversaries [37], [38], [39]. Most raters are honest, and they follow the protocol by faithfully submitting ratings according to the received products or services [40], [41]. A small part of raters are malicious, and they may launch multiple rating attacks and abnormal rating attacks. The multiple rating attack refers to a rater's submitting more than one ratings to a ratee after just one financial transaction between them. The abnormal rating attack refers to a rater's submitting to the ratee a falsified rating that is out of the normal range. Ratees strictly follow the protocol, but they are curious about the identities and ratings of raters in rating transactions. E-commerce platforms have the same security assumptions as of the ratees, and they try to learn the ratings of raters in rating transactions. The CA is fully trusted, and adversaries cannot breach it [42], [43]. External adversaries can eavesdrop on communication channels in an attempt to violate the privacy of customers.

Trusted identity information of the different actors is established in the beginning of the blockchain and stored in all blockchain nodes. Management of keys for the proposed system is performed by the certificate authority. Ethereum Name Service (ENS) is not needed in our implementation as we used one public cloud, two laptops, and one desktop to be four nodes in a local and small blockchain network. It is actually a local network and the four nodes only need an IP address to "find" each other, thus the ENS, i.e., address mapping, is not needed. Generally, ENS is used in real-world scenario with a large number of nodes.

C. Design Objectives

Our design objectives are to propose a privacy-preserving reputation scheme. Specifically, the following three goals should be achieved:

Privacy. (1) Identity Privacy. Other entities cannot identify the real identity of a unique customer who produces a particular rating. (2) Rating Privacy. The rating in each customer's rating transaction should be hidden from other entities. (3) Unlinkability. Two credentials of the same customer cannot be correlated, i.e., they cannot be linked any better than guessing even if they are from the same customer.

Security. The reputation system must resist the two attacks, i.e., multiple rating attack and abnormal rating attack. The customers are only allowed to submit one rating transaction toward a supplier after one financial transaction and abnormal ratings in rating transactions will be detected.

Efficiency. (1) The computational costs in the mining and verifying rating transactions should be lightweight. (2) The total length of a rating transaction should be as short as possible.

D. Technical Challenges

Challenge 1: Preserving customers' anonymity while defending against multiple rating attack under the blockchain-based framework. As mentioned above, customers will not engage in rating activity if their privacy is not preserved. This privacy issue becomes worse in a blockchain network. Specifically, a customer must submit a rating transaction to give a numerical comment towards the supplier. The rating transaction could potentially contain the identification information of the customer. Anonymity is to protect the identity of the customer. However, multiple ratings (from one customer toward one supplier after one financial transaction) should be detected if the anonymous customer's rating is out of the normal range.

Challenge 2: Leveraging the blockchain technology to realize the reputation system among E-commerce platforms while

lacking an effective approach to resist abnormal rating attack. The consortium blockchain has offered a promising way to solve the trust issue in reputation system among multiple E-commerce platforms. A simple solution may be achieved by asking the customers to submit a rating into a smart contract and asking nodes to update the reputation for suppliers. However, this approach still fails due to the abnormal ratings from malicious customers. Therefore, it is nontrivial to design a correctness protocol using smart contracts that can verify the encrypted ratings without causing too much computational costs.

IV. Preliminaries

In this section, we briefly revisit blind signature [22], zero-knowledge range proof [23], secure multiparty computation [26], [24], blockchain [44], [45], [46], and consensus hashing [27].

A. Blind Signature

A blind signature scheme [22] allows a user to obtain a signature on a message which the signer does not know. It provides blindness and unforgeability. It consists of five algorithms:

$\mathsf{BGGen}_{\mathcal{R}}(1^{\kappa})$: Generate a Type-3 bilinear group $\mathrm{G} = (\hat{p}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2)$ with order p of length κ, two generators g_1 and g_2 for $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively, and a bilinear pairing (map) $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$.

$\mathsf{Kengen}_{\mathcal{R}}(\mathrm{G}, \ell)$: Given G and a length $\ell > 1$, choose $(x_i)_{i\in[\ell]} \xleftarrow{R} (\mathbb{Z}^*_{\hat{p}})^{\ell}$, set $\mathsf{sk} \leftarrow (x_i)_{i\in[\ell]}, \mathsf{pk}_{\mathcal{R}} \leftarrow (\hat{X}_i)_{i\in[\ell]} = (x_i g_2)_{i\in[\ell]}$ and output $(\mathsf{sk}, \mathsf{pk}_{\mathcal{R}})$.

$\mathsf{Sign}(M, \mathsf{sk})$: Given a message $M = (M_i)_{i\in[\ell]} \in (\mathbb{G}^*_1)^{\ell}$ and a secret key $\mathsf{sk} = (x_i)_{i\in[\ell]}$, choose $y \xleftarrow{R} \mathbb{Z}^*_{\hat{p}}$ and output $\sigma = (A, B, \hat{B})$ with $A \xleftarrow{R} y \sum_{i\in[\ell]} x_i M_i$, $B \xleftarrow{R} \frac{1}{y} g_1$, and $\hat{B} \xleftarrow{R} \frac{1}{y} g_2$.

$\mathsf{Verify}_{\mathcal{R}}(M, \sigma, \mathsf{pk}_{\mathcal{R}})$: Given $M, \sigma = (A, B, \hat{B})$, and a public key $\mathsf{pk}_{\mathcal{R}}$, output 1 if $\prod_{i\in[\ell]} e(M_i, \hat{X}_i) = e(A, \hat{B})$ and $e(B, g_2) = e(g_1, \hat{B})$, and output 0 otherwise.

$\mathsf{ChaRep}_{\mathcal{R}}(M, \sigma, \theta, \mathsf{pk}_{\mathcal{R}})$: Given a message M and $\sigma = (A, B, \hat{B})$, choose $\theta \in \mathbb{Z}^*_{\hat{p}}$ and $\mathsf{pk}_{\mathcal{R}}$, output $\perp$ if $\mathsf{Verify}_{\mathcal{R}}(M, \sigma, \mathsf{pk}_{\mathcal{R}}) = 0$. Otherwise, pick $\eta \xleftarrow{R} \mathbb{Z}^*_{\hat{p}}$ and output $(\theta M, \sigma')$ with $\sigma' \leftarrow (\theta\eta A, \frac{1}{\eta} B, \frac{1}{\eta} \hat{B})$.

B. Zero-knowledge Range Proof

A zero-knowledge range proof scheme [23] allows a prover $\mathcal{P}$ to convince a verifier $\mathcal{V}$ that a committed value is within a given range $\Phi = [LB, UB]$. It is composed of five steps: $\mathcal{V}$ chooses $x \in_R \mathbb{Z}_{\hat{p}}$ and computes $X \leftarrow g^x$ and $X_i \leftarrow g^{\frac{1}{x+i}}$ for each $i \in \Phi$ where g is an element of $\mathbb{G}_1 =< g >=< h >$ there are discrete and countable items in this range. $\mathcal{V}$ sends X and $\{X_i\}$ to $\mathcal{P}$. For a number m, $\mathcal{P}$ chooses $o \in_R \mathbb{Z}_{\hat{p}}$, computes $O \leftarrow X^o_m$; $\mathcal{P}$ chooses $\rho_1, \rho_2, \rho_3 \in_R \mathbb{Z}_{\hat{p}}$, computes $C = g^m h^r, D_1 \leftarrow e(O, g)^{-\rho_1} e(g, g)^{\rho_2}$ and $D_2 \leftarrow g^{\rho_1} h^{\rho_3}$, and sends O, C, D_1, and D_2 to $\mathcal{V}$. $\mathcal{V}$ sends a random challenge $c \in_R \mathbb{Z}_{\hat{p}}$ to $\mathcal{P}$. $\mathcal{P}$ sends $z_1 \leftarrow \rho_1 - mc, z_2 \leftarrow \rho_2 - oc$, and $z_3 \leftarrow \rho_3 - rc$ to V. $\mathcal{V}$ verifies whether $D_2 = C^c h^{\rho_3} g^{\rho_1}$ and $D_1 = e(O, X)^c e(O, g)^{-\rho_1} e(g, g)^{\rho_2}$. In this work, the customer is the prover. Suppliers and platforms are the verifiers.

C. Secure Multiparty Computation

Secure multiparty computation enables multiple parties to compute an output based on the "encrypted" inputs of the parties while the values of their inputs are kept secret. A secure sum protocol [24] is based on the threshold Paillier cryptosystem [26]. It can calculate the sum sum of n users' data without exposing them, and the decryption of sum ciphertext sc only needs t $(1 < t < n)$ users. Specifically, the secure sum protocol consists of four algorithms:

$\mathsf{KeyGen}(1^{\kappa})$: given a security parameter 1^{κ}, choose two prime numbers p and q, compute $n = pq$ and a generator $\tilde{g}$, choose random numbers a_i $(0 < i < z)$, make a polynomial $f(x)$, set the public key as n, and compute user i's secret share sh_i. We note that the share is not the shared file among all nodes, but a share of the secret key.

$\mathsf{Enc}(m, n)$: given a message m, pick a random number r, and compute a ciphertext ct.

$\mathsf{ShareDec}(sc, ss_i)$: given a ciphertext sc, user i computes a share decryption sd_i using secret share ss_i and a zero-knowledge proof pr_i.

$\mathsf{ShareCom}(pd_1, ..., pd_z, n)$: given z share decryptions pd_1, ..., pd_z, multiply them to obtain sc and recover m.

D. Blockchain

A typical blockchain is a publicly shared and commonly maintained digital ledger. Everyone can join in and competes in mining new blocks in order to receive rewards. A CBC is a permissioned blockchain running among identified entities, and it protects transactions between users who do not fully trust each other [44]. Specifically, different users agree on collaboration and co-construct a consortium blockchain CBC. Such CBC validates all internal transactions sent by users. The users select a winning user according to a predefined group consensus algorithm in each period and this elected user packs a new data block. The new block is cryptographically chained to the last block. CBC is already applied in vehicular networks [45] and grid networks [46].

E. Consensus Hashing

Consensus hashing is an efficient hashing technique that helps verify data items along a chain of multiple data items. Each consensus hash $CH(di)$ is computed from a specific series of prior consensus hashes P_{di} : $CH(di) = hash(\mathcal{P}_{di})$ where $\mathcal{P}_{di} = \{CH(di - 2^i) | i \in \mathbb{N}, di - 2^i \geq di_0\}$ and di_0 is the initial data item. If the final consensus hash matches the trusted consensus hash at di_n, then the database associated with di_n is trustworthy and the data item can start processing data items after di_n. The construction of CH allows a user to verify the authenticity of any data item from a data item with a height $di_{prior} < di$, using only a logarithmic number of queries.

V. The Proposed System: RepChain

In this section, we present RepChain which has five phases: initializing system, registering entities, accessing and rating reputation, processing shares, maintaining blockchain and updating reputation, and rating verification. The state machine model for rating process is shown in Fig. 3. Rating Request originates in accessing and rating reputation, Share Processing and Sum Computing correspond to processing shares, Reputation Updating is completed in updating reputation, and finally Completed resides in rating verification. We provide an overview of the RepChain by using Algorithm 1.

Here, we give an example to explain the different steps of the system. A customer Alice has completed a financial transaction with a supplier Bob. Next, Alice sends a rating request to the blockchain network. The blockchain nodes perform share processing and upload a share transaction. The winning node helps Bob update the reputation value by collecting share decryptions and uploading an update transaction. Finally, a blockchain node verifies the corresponds transactions and sends a complete transaction to the blockchain network.

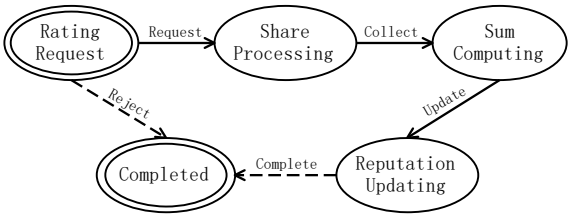


Fig. 3. The State Machine Model for Rating Process.

Algorithm 1: RepChain

Input: Security parameter.

Output: System parameters, registration information, rating transaction, share transaction, new block, new reputation.

/*Initializing system*/

1. CA generates CA generates system parameters;

/*Registration Entities*/

2. A supplier S_i registers to obtain registration information;

3. A platform P_j registers to obtain registration information;

/*Accessing and Rating Reputation*/

4. A customer C_i accesses reputation;

5. C_i conduct a financial transaction with a supplier S_j and rates S_j with a rating transaction;

/*Processing Shares, Maintaining Blockchain and Updating Reputation*/

6. Each platform P_k processes a share and sends a share transaction;

7. A new block is created;

8. Suppliers' reputations are updated;

/*Rating Verification*/

9. Rating transactions are verified.

A. Initializing System

The CA initializes the whole reputation system as follows.

- Given a security parameter κ, CA generates three cyclic groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ of order $\hat{p}$ with $\log_2 \hat{p} = \kappa$, a generator g_1 for $\mathbb{G}_1$, a generator g_2 for $\mathbb{G}_2$, and a bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$.
- CA initiates a range $[LB, UB]$ and there are 11 items, i.e., $\{0, 1, 2, ..., 9, 10\}$ in this range.
- CA chooses two prime numbers p and q that satisfies $p = 2p' + 1, q = 2q' + 1$ where p', q' are prime numbers, computes $n = pq, n' = p'q'$ and a generator $\tilde{g}$, decides on sp to determine plaintext space n^{sp}, picks d satisfying $d = 0 \bmod n'$, and $d = 1 \bmod n^{sp}$, chooses random numbers a_i $(0 < i < n_3)$, make a polynomial $f(x) = \sum_{i=0}^{n_3-1} a_i x^i \bmod n^{sp} n'$ and $a_0 = d$, and sets the public key as n.
- CA publishes public parameters $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{p}, g_1, g_2, e, LB, UB, p, q, \tilde{g}, n, sp)$.

Finally, CBC is initialized as follows.

- Initialization. Split time to a sequence of slots $\{ts_1, ts_2, ...\}$. Each platform has a synchronized clock indicating when to execute a distributed protocol and append block in the current time slot. A leader election function $\mathsf{F}(\cdot)$ is assigned to each platform and a rating transaction pool RTP is initialized as empty.
- Stage One. A default stake distribution is included in the genesis block B_0 including an empty blockheader, platforms' identities $\{P_i\}_{i=1}^{n_3}$, public addresses $\{pk_{P_i}\}_{i=1}^{n_2}$, a stake list, i.e., a list of reputation values $\{rep_{P_i}\}_{i=1}^{n_2}$ of platforms, and signatures of platforms. Each platform P_i sets a local ledger $CB_i = B_0$.

B. Registration Entities

A supplier S_i registers to CA as follows.

- Given public parameters and a key vector length $\ell = 2$, CA chooses $(x_i)_{i\in[\ell]} \xleftarrow{R} (\mathbb{Z}_{\hat{p}}^*)^\ell$, sets $\mathsf{sk} \leftarrow (x_i)_{i\in[\ell]}$ and $\mathsf{pk}_{\mathcal{R}} \leftarrow (\hat{X}_i)_{i\in[\ell]} = (x_i g_2)_{i\in[\ell]}$.
- Given a rating vector length $\ell' = \lceil UB - LB \rceil = 7$, CA picks $\acute{q} \xleftarrow{R} \mathbb{Z}_{\hat{p}}^*$ and $(\acute{p}_i)_{i\in[\ell']} \xleftarrow{R} (\mathbb{Z}_p^*)^{\ell'}$, and sets $Q_1 \leftarrow \acute{q} g, Q_2 \leftarrow \acute{q} g_2$, and $(g_i)_{i\in[\ell']} \xleftarrow{R} (\acute{p}_i g)_{i\in[\ell']}$. CA returns $(\mathsf{sk}, \mathsf{pk} = (\mathsf{pk}_{\mathcal{R}}, (g_i)_{i\in[\ell']}, Q_1, Q_2))$ to S_i.

Now A platform P_j registers to CA. CA computes a secret share $ss_j = f(j)$, a verification key $vk_j = v^{\Delta ss_j}$ and $\Delta = n_1!$, and returns (d, ss_j, vk_j) to P_k.

C. Accessing and Rating Reputation

1) Accessing Reputation: Before initiating a financial transaction, a customer C_i could access the reputation of a supplier S_j. C_i looks up S_j's reputation rep_{S_j} by putting S_j's public key pk_{S_j} to a CBC interface and receives a query result.

2) Rating Reputation: If C_i is assured of this supplier, she/he will proceed to conduct a financial transaction with S_j, e.g., purchase a camera or hail for a ride. After the financial transaction is complete, the customer interacts with the supplier to generate a rating transaction as follows.

– S_j chooses $x \in_R \mathbb{Z}_{\hat{p}}$, computes $X \leftarrow g^x$ and $X_i \leftarrow g^{\frac{1}{x+i}}$ for each $i \in \Phi$, and sends X and $\{X_i\}$ to C_i. The computation operations of this step could be reduced using pre-computation.
– C_i generates a rating rt_{ij} for S_j where rt_{ij} should be a valid real number belonging to $[LB, UB]$, but a malicious customer can produce a number outside this range.
– C_i chooses $o \in_R \mathbb{Z}_{\hat{p}}$, computes $O \leftarrow X^o_{rt_{ij}}$, choose $\rho_1, \rho_2, \rho_3 \in_R \mathbb{Z}_{\hat{p}}$, computes $C_{ij} = g^{rt_{ij}} h^r, D^1_{ij} \leftarrow e(O,g)^{-\rho_1} e(g,g)^{\rho_2}$ and $D^2_{ij} \leftarrow g^{\rho_1} h^{\rho_3}$, computes $z^1_{ij} \leftarrow \rho_1 - rt_{ij} c_{ij}, z^2_{ij} \leftarrow \rho_2 - o c_{ij}, z^3_{ij} \leftarrow \rho_3 - r c_{ij}$, and $c_{ij} = H(O_{ij}, C_{ij}, D^1_{ij}, D^2_{ij}, z^1_{ij}, z^2_{ij}, z^3_{ij})$. In this way, C_i has produced a range proof pf^1_{ij} for rt_{ij}.
– C_i transforms rt_{ij} into a vector $\mathbf{rt}^b_{ij}$ with $b \in [\ell']$, chooses $s_1 \xleftarrow{R} \mathbb{Z}^*_{\hat{p}}$ and $s_2 \xleftarrow{R} \mathbb{Z}^*_{\hat{p}}$ such that $\sum_{b\in[\ell']} rt^w_{ij} g_b + s_2 Q_1 \neq 0_{\mathbb{G}_1}$, and computes a commitment:

$$Com_i = \sum_{b\in[\ell']} rt^w_{ij} g_b + s_2 Q_1. \tag{1}$$

– C_i computes $RT = (s_1 Com_i, s_1 g) \in (\mathbb{G}^*_1)^2$ and sends to S_j $((C_{ij}, O_{ij}, D^1_{ij}, D^2_{ij}, z^1_{ij}, z^2_{ij}, z^3_{ij}, c_{ij}), RT)$ with a proof pf^2_{ij} that RT commits to $\mathbf{rt}^b_{ij}$:

$$\mathsf{PoK}\left\{ ((\alpha_z)_{z\in U}, \beta, \gamma): \begin{array}{r} s_1 Com_i = \sum_{b\in\ell'} rt^w_{ij} g_i \\ + \sum_{z\in U} \alpha_z H_z + \beta H_{Q_1} \wedge \\ \bigwedge_{i\in[\ell']} (H_i = \gamma g_i) \wedge \\ H_{Q_1} = \gamma Q_1 \wedge \\ s_1 g = \gamma g \end{array} \right\}.$$

S_j verifies whether $D^2_{ij} = C^{c_{ij}}_{ij} h^{\rho_3} g^{\rho_1}$ and $D^1_{ij} = e(O_{ij}, X)^{c_{ij}} e(O,g)^{-\rho_1} e(g,g)^{\rho_2}$. If either of them does not hold, the corresponding rating rt_{ij} is abnormal, S_j refuses to help C_i generate a one-time rating credential and drops the rating. Otherwise, S_j computes a signature and returns it to C_i:

$$\sigma_{ji} = (A, B, \hat{B}) = (y \sum_{i\in[\ell]} x_i RT_i, \frac{1}{y} g_1, \frac{1}{y} g_2). \tag{2}$$

Next, C_i prepares a rating transaction as follows.

– C_i verifies whether $\prod_{i\in[\ell]} e(RT_i, \hat{X}_i) = e(A, \hat{B})$ and $e(B, g_2) = e(g, \hat{B})$. It they hold, C_i picks $\eta \xleftarrow{R} \mathbb{Z}^*_{\hat{p}}$ and computes:

$$((\frac{1}{s_1} RT), \sigma'_i) = ((Com_i, g), \frac{\eta}{s_1} A, \frac{1}{s_1} B, \frac{1}{s_1} \hat{B}). \tag{3}$$

By doing so, C_i has held a one-time rating credential:

$$rc_i = (Com_i, \sigma'_i, s_2). \tag{4}$$

– C_i picks $r \xleftarrow{R} \mathbb{Z}^*_{n^{sp+1}}$ and encrypts rt_{ij}:

$$ct_{ij} = \tilde{g}^{rt_{ij}} r^{n^{sp}} \bmod n^{sp+1}. \tag{5}$$

– C_i broadcasts to the rating transaction pool RTP a rating request, i.e. a rating transaction:

$$Tx_{ij} = (S_j, ct_{ij}, rc_i, H_{ij}). \tag{6}$$

where H_{ij} is the unique identifier of the request.

D. Processing Shares, Maintaining Blockchain and Updating Reputation

1) Processing Shares: Each node P_k computes a share decryption sd_{kj} for supplier S_j as follows.

– P_k computes a share decryption $sd_{kj} = ct_j^{2\Delta ss_k}$, along with a zero-knowledge proof (ZKP) that $log_{ct_j^4}(sd^2_{kj}) = log_v(vk_k)$ where ct_j is the product of all ratings for S_j.
– P_k generates a signature σ_{kj} on sd_{kj} and sends a share transaction $Tx_{P_k} = (sd_{kj}, \sigma_{kj})$ to DP.

2) Maintaining Blockchain: In each time slot ts_i, one platform $\hat{P}_j$ is elected to create a new block with a probability $\Pr_{\hat{P}_j}$ of being elected is proportional to its stake.

– Each platform runs the leader election function F [35] which takes inputs $\{pk_{P_o}\}^{n_3}_{o=1}$, $\{rep_{P_o}\}^{n_3}_{o=1}$, $|\{sd^{ts_i}_{kj}\}|$ and ts_i, and outputs a winning $\hat{P}_i$, where $|\{sd^{ts_i}_{kj}\}|$ is the total number of share decryptions of $\hat{P}_j$ in ts_i, and $\Pr_{P_o} = rep_{P_o} / \sum^{n_3}_{j=1} rep_{P_j}$.
– The elected node $\hat{P}_i$ verifies n_{Tx} transactions and attaches a signature. If the verification passes, $\hat{P}_i$ creates a new block B_{ts_i} containing a blockheader BH_{ts_i} (including a block number bn_{ts_i}, a hash of the previous blockheader $H(B_{ts_{i-1}})$, a Merkle hash root MHR_{ts_i} of Merkle tree constructed from n_{Tx} transactions, a timestamp T_{ts_i}, platforms' updated stakes rep_{sl_i}, and a signature $\sigma^{ts_i}_{\hat{P}_i}$).
– $\hat{P}_i$ calculates a consensus hash for each string of rating reputations for efficient verification which we will explain in Section 5.5.
– Then it adds B_{ts_i} to CBC and broadcasts it to the blockchain network.

3) Updating Reputation: After collecting N share decryptions for supplier S_j from N nodes, S_j updates his reputation values as follows.

– $\hat{P}$ computes a share combining $sc_{kj} = \prod_{i\in\mathcal{S}} sd_i^{2\lambda^{\mathcal{S}}_i} \bmod n^{sp+1}$ where $\lambda^{\mathcal{S}}_i = \prod_{j\in\mathcal{S}\wedge j\neq i} \frac{-i}{i-j} \in Z$.
– $\hat{P}$ computes a sum of ratings sum_j for S_j and sends an update transaction $Tx^{S_j}_{\hat{P}} = (sum_j, \sigma^{S_j}_{\hat{P}})$ to S_j, given that sc_{kj} has the form $sc_{kj} = CT^{4\Delta^2 f(0)}$ where CT is the encryption of sum_j. Since $4\Delta^2 d = 0 \bmod \lambda = 0$ and $4\Delta^2 d = 4\Delta^2 \bmod n^{sp}$ where λ is the least common multiple of $p-1$ and $q-1$, then $sc_{kj} = (1+n)^{4\Delta^2 sum_j} \bmod n^{sp+1}$. Next, sum_j could be extract part by part [26].
– S_j updates reputation and number of transactions as follows: $rep^{new}_j = (rep^{old}_j * num_j + sum_{kj})/(num_j + N), num_j = num_j + N$ where rep^{new}_j and rep^{old}_j are S_j's new and current reputation, and num_j is S_j's number of previously received ratings.

E. Rating Verification

Given a chain of rating transactions, it should be efficient to verify their authenticity. Here, we utilize the consensus hashing [27] to achieve this goal.

– Each rating transaction is already appended with a hash value, and the elected nodes add a consensus hash in each string of rating transaction.

– For instance, say we have a string of rating transactions $\mathcal{T}_i = \{Tx_{ij}\}_{j=1}^{N'}$ for S_i which are verified and waiting to be added in the next block.
– The elected platform $\hat{P}$ starts with the first rating transaction Tx_{i1} and calculates a consensus hash $CH_{ij} = H(\mathcal{P}_{ij})$ for each Tx_{ij}, where $\mathcal{P}_{ij} = \{CH_{j-2^o} | o \in \mathbb{N}, j - 2^o \geq Height\{Tx_j\}\}$, Tx_j is the first rating transaction for S_j and $Height$ is the height of the transaction.

TABLE III
REPCHAIN CONTRACT

On receiving ("**Request**", $Tx_{ij} = (S_j, ct_{ij}, rc_i)$) from C_i
Create CurrentState = Rating Request;
Insert rating transaction pool $RTP[H(Tx_{P_k})] = H(Tx_{P_k})$;
Initialize $RTP[H(Tx_{P_k})].shareNum = 0$;
Set CurrentState = Share Processing;
Broadcast ("A rating request has been created.);
On receiving ("**Collect**", $Tx_{P_k} = (sd_{kj}, \sigma_{kj})$ from P_k
Run the leader election function F) and create a new block;
Await N share decryptions from nodes;
Set $RTP[H(Tx_{P_k})].shareNum = N$;
Set CurrentState = Sum Computing;
Broadcast ("Enough share decryptions have been collected.);
On receiving ("**Update**", $Tx_{\hat{P}}^{S_j} = (sum_j, \sigma_{\hat{P}}^{S_j})$ from $\hat{P}$
Set $RTP[H(Tx_{P_k})].newRatings = (sum_j)$;
Set CurrentState = Reputation Updating;
Broadcast ("The sum of new ratings has been calculated.);
On receiving ("**Reject**", Tx_{ij}) from P_i
Delete $RTP[H(Tx_{P_k})]$;
Set CurrentState = Completed;
Broadcast ("The rating process has been rejected.);
On receiving ("**Complete**", Tx_{tm})
Verify signature;
Set CurrentState = Completed;
Broadcast ("A rating request has been completed.);

An example of a fast verification of rating transactions is shown in Fig. 4. Specifically, we now have 10 ratings for supplier Alex from 10 different customers, and the previously elected platforms have calculated their consensus hashes that are stored on the CBC. Assume we want to check the validity of rating transaction $Tx3$ and we start from the current rating transaction Tx. We verify Tx with $Tx1, Tx2, Tx4$, and $Tx8$ by checking $CH_{Tx} \stackrel{?}{=} H(CH_{Tx_1}||CH_{Tx_2}||CH_{Tx_4}||CH_{Tx_8})$. Next we can proceed to verify $Tx2$ with $Tx3, Tx4$, and $Tx6$. Finally, we verify $Tx3$ by checking $CH_{Tx3} \stackrel{?}{=} H(CH_{Tx_4}||CH_{Tx_5}||CH_{Tx_7})$.

If verification succeeds, the rating process instance is terminated by sending a complete transaction Tx_{tm}. We present the whole state machine transition for the reputation system as the RepChain contract in Table III. Rating Request includes the first function in the smart contract, Share Processing and Sum Computing include the second function, Reputation Updating includes the third function, Completed refers to the last function. If verification in any phase fails, the fourth function will be invoked.

VI. SECURITY AND PRIVACY ANALYSIS

Rating Privacy requires that the rating in each customer's rating transaction should be hidden from other entities.

Theorem 1: The rating privacy is protected if the decisional composite residuosity assumption (DCRA) is true, i.e., the rating encryption scheme is indistinguishable encryptions under a chosen-plaintext attack (CPA) if for all probabilistic polynomial-time (PPT) adversaries $\mathcal{A}$, the $\mathsf{negl}(\kappa)$ is a negligible function in the following inequation:

$$\Pr[\mathsf{PubK}^{\text{cpa}}_{\mathcal{A},\prod} = 1] \leq \tfrac{1}{2} + \text{negl}(\kappa).$$

Proof: Let Π be the encryption scheme in the RepChain and $\mathcal{A}$ be a PPT adversary aiming to attack Π with $Q = Q(kappa)$ an upper bound on the number of queries that A makes to an oracle $\mathcal{O}$. Since the message space is $Z_{n^{sp}}$, thereby, a message m can be expressed in a u-tuple $(m_u, m_{u-1}, ..., m_1)$ where each $m_i \in Z_n$ and $m = \sum_{i=o}^{u-1} m_{i+1} n^i$. Now we show how to construct an adversary $\mathcal{A}'$ that runs $\mathcal{A}$ and aims to attack Π'. The idea is to assume that Π is not secure, then we can build a reduction showing how to transform $\mathcal{A}$ into an efficient algorithm $\mathcal{A}'$ that solves the underlying hard problem, i.e., DCRA.

Algorithm $\mathcal{A}'$:

1. $\mathcal{A}'$ is given public parameters (p, q, g, n, sp) in Π and access to an encryption oracle $\mathcal{O}$.

2. $\mathcal{A}'$ runs $\mathcal{A}$, answering a oracle query $(\hat{m}_0, \hat{m}_1)$ with a challenge ciphertext c_b as follows:

2.a. For m_0 in the form of m_u, $\mathcal{A}'$, generates $(m_{u-1}, m_{u-2}, ..., m_1)$, computes $c = Enc_n(m_{u-1}, m_{u-2}, ..., m_1)$, and returns $c_b c$ to $\mathcal{A}$.

2.b. For m_0 in the form of m_1, $\mathcal{A}'$, generates $(m_u, m_{u-1}, ..., m_2)$, computes $c = Enc_n(m_u, m_{u-1}, ..., m_2)$, and returns $c_b c$ to $\mathcal{A}$.

2.c. For m_0 in the form of $m_i (u < i < 1)$, $\mathcal{A}'$, generates $(m_u, ..., m_{i+1}, m_{i-1}, ..., m_1)$ computes $c_1 = Enc_n(m_u, m_{u-1}, ..., m_{i+1})$ and $c_2 = Enc_n(m_{i-1}, m_{i-2}, ..., m_1)$, and returns $c_1 c_b c_2$ to $\mathcal{A}$.

3. $\mathcal{A}'$ outputs what $\mathcal{A}$ outputs.

The view of $\mathcal{A}$ when run as a subroutine by $\mathcal{A}'$ in the above experiment $\mathsf{PubK}^{\text{cpa}}_{\mathcal{A},\prod'}$ is identical to the view of $\mathcal{A}$ in experiment $\mathsf{PubK}^{\text{cpa}}_{\mathcal{A},\prod}$. Since

$$\begin{aligned}
&\Pr[\mathsf{PubK}^{\text{cpa}}_{\mathcal{A},\prod'} = 1] \\
&= \tfrac{1}{2}\Pr[\mathsf{PubK}^{\text{cpa}}_{\mathcal{A},\prod'} = 1 | b = 0] + \tfrac{1}{2}\Pr[\mathsf{PubK}^{\text{cpa}}_{\mathcal{A},\prod'} = 1 | b = 1] \\
&= \tfrac{1}{2}\Pr[\mathsf{PubK}^{\text{cpa}}_{\mathcal{A},\prod} = 1 | b = 0] + \tfrac{1}{2}\Pr[\mathsf{PubK}^{\text{cpa}}_{\mathcal{A},\prod} = 1 | b = 1] \\
&= \tfrac{1}{2}\Pr[\mathsf{PubK}^{\text{cpa}}_{\mathcal{A},\prod'} = 1] \\
&= \tfrac{1}{2} + \text{negl}(\kappa).
\end{aligned}$$

Therefore, if $\mathcal{A}$ succeeds in breaking the rating encryption scheme, then $\mathcal{A}'$ can break the underlying DCRA with a non-negligible probability.□

Identity Privacy requires that other entities cannot identify the real identity of a unique customer who produces a particular rating. Given a secure commitment scheme together with a blind signature scheme with attributes implies a one-show credential system [22], where each user holding a credential and some attributes proves its qualification without revealing undisclosed attributes. No entity can link a credential to its holder, but different appearances of the same credential are linkable. The blindness of the signature scheme ensures that, given two signatures generated on commitments of his own choice, the signer, as well as other entities, cannot link a

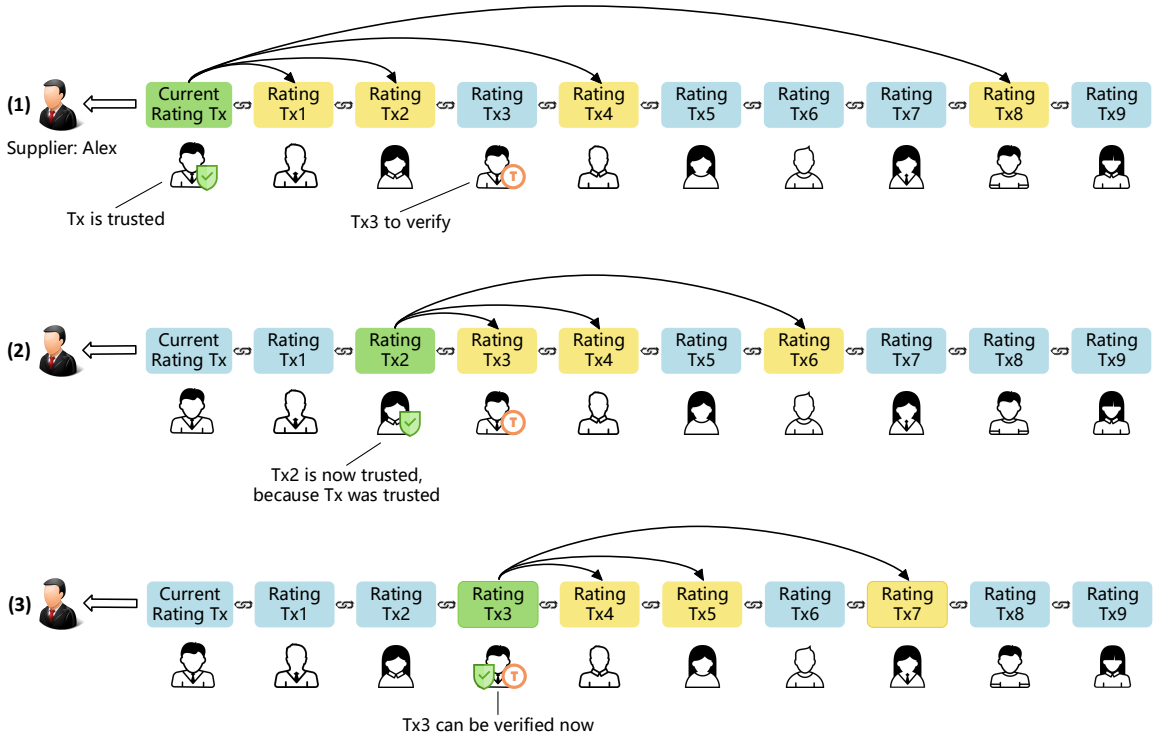


Fig. 4. Fast Verification of Rating Transactions

signature to its issuing. Hence, RepChain provides identity privacy.

Unlinkability states that other entities are not able to correlate a customer's two credentials. Two credentials cannot be linked any better than guessing even if they are from the same customer.

In registration, a customer C_i registers a generalized Pedersen commitment Com_i to his/her vector (attribute) $\mathbf{rt}_{ij}^b$. In the preparation and validation phase, the customer engages in a blind-signature-with-attributes protocol for a rating rt_{ij} and another combined commitment Com_i'. Finally, the credential is the customer output of a blind-signature-with-attributes protocol resulting in a signature on rating rt_{ij} and a blinded Pedersen commitment Com_i''. The latter contains the same vectors as Com_i, but is unlinkable to Com_i and Com_i' [22].

Resistance to multiple rating attack. The reputation system must resist the attack, i.e., customers are only allowed to submit one rating transaction toward a supplier after one financial transaction.

Given a secure commitment scheme together with a blind signature scheme with attributes implies a one-show credential system [47]. Such one-show anonymous credentials guarantee that a credential can only be once. If a credential is used more than once, the customer's identity will be discovered. Hence, RepChain is resistant to multiple rating attack.

Resistance to abnormal rating attack. The reputation system must resist the attack, i.e., the customer cannot submit abnormal ratings in rating transactions.

We utilize the range proof [23] to check whether a rating belongs to the legal range $[LB, UB]$. If a rating rt_{ij} is out of the range, we affirm that the rating is abnormal. Specifically, each customer C_i generates a proof $(O_{ij}, C_{ij}, D_{ij}^1, D_{ij}^2, z_{ij}^1, z_{ij}^2, z_{ij}^3, c_{ij})$ to prove that $rt_{ij} \in [LB, UB]$ without disclosing rt_{ij} to other entities. Supplier S_j checks the proof to decide whether the underlying rating belongs to the range.

Theorem 2: If the $|\Phi|$-Strong Diffie-Hellman assumption holds, then RepChain is a zero-knowledge argument of set membership for a set $|\Phi|$, i.e., resistant to abnormal rating attack.

Proof: The extraction property means for any prover that convinces verifier with probability ϵ, there exists an extractor $\mathcal{E}$ which interacts with the prover and outputs a witness (w_1, w_2, w_3) with probability $poly(\epsilon)$. Furthermore, if we assume that the extractor has two transcripts, i.e., $(X, X_i, C_{ij}, O_{ij}, D_{ij}^1, D_{ij}^2, z_{ij}^1, z_{ij}^{1'}, z_{ij}^2, z_{ij}^{2'}, z_{ij}^3, z_{ij}^{3'}, c_{ij}, c_{ij}^{'})$, we can compute the following equations to obtain the witness:

$$w_1 = \frac{z_{ij}^1 - z_{ij}^{1'}}{c_{ij}^{'} - c_{ij}}, w_2 = \frac{z_{ij}^2 - z_{ij}^{2'}}{c_{ij}^{'} - c_{ij}}, w_1 = \frac{z_{ij}^3 - z_{ij}^{3'}}{c_{ij}^{'} - c_{ij}}$$

$\mathcal{E}$ succeeds when $c_{ij}^{'} - c_{ij}$ is invertible. If $w_1 \notin \Phi$, then the prover can be used to launch a weak chosen-message attack against the BBS scheme with successful probability $poly(\epsilon)$. Therefore, ϵ must be negligible.

A few discussions. We do not disclose the identity of raters who provides an out of bound value but we only discard the ratings. We do reveal the identity of malicious raters who launches a multiple rating attack. After the revelation, we enforce minor punishment on the rater within the system. For example, punishment includes reducing account credibility, revealing a part of the identity, etc. Meanwhile, revealing

the identity is an optional choice depending on different applications.

Since we use privacy-preserving ratings, thus rating verifiers and ratees cannot see the ratings in plaintext. Therefore, we need to locate these abnormal packets by using zero knowledge proofs. If the proof does not hold, we will discard the corresponding rating. Other techniques [48], [49] can be used to make the scheme more efficient and even process them in batches. Moreover, we do not consider the scenario where a developer or network error generates the out of bounds ratings.

Smart Contract often deals with high-value assets and has been a target of security attacks [50], [51]. It is also challenging to write smart contracts that are free of vulnerabilities. A well-known attack has caused the loss of 3.6 million Ethers when the smart contract allowed an attacker to recursively call a function before the initial call was completed [52]. To improve the security of smart contracts, we could resort to safety verification [53] and safety verifier [54].

VII. Performance Analysis

In this section, we implement a prototype of RepChain and analyze its computational costs, communication overhead, and monetary cost.

A. Experiment Settings

We use Ethereum blockchain test platform. We instantiate four nodes: one node with a public IP address as the boot node. The other three nodes are deployed on two laptops and one desktop. The consensus mechanism is Clique (Proof-of-Authority (PoA)). We first installed Ethereum-Wallet [55] and geth [56], then created a genesis block, and initiate the consortium blockchain. We set the block time as 10 seconds. We use JPBC library to implement cryptographic primitives with an elliptic curve being defined as $y^2 = x^3 + x$ over $\mathbb{F}_{q0}$ [57]. The rating transactions from customers to suppliers are randomly generated, and the number of required share decryptions (i.e., ratings) for one update is $N = 5$. The key experimental parameters are listed in Table IV.

TABLE IV
Key Experimental Parameters

Parameters	Value
$\lvert\hat{p}\rvert, \lvert q_0\rvert, sp, \lvert n\rvert$	$160, 512, 10, 1024$
n_1, n_2, n_3, n_{Tx}, N	$100, 100, 4, 20, 5$
H	SHA256

B. Computational Costs

We now analyze the computational costs for customers, suppliers, platforms, and rating verifier by counting the number of cryptographic operations. We define $Mu_1/Mu_2/Mu_T$, $Ad_1/Ad_2/Ad_T$, Di_T/Ex_T, Ex/Mu, BP, H, and $Add/Sub/Mul/Div/Exp$ as the operation of multiplication in $\mathbb{G}_1/\mathbb{G}_2/\mathbb{G}_T$, addition in $\mathbb{G}_1/\mathbb{G}_2/\mathbb{G}_T$, division/exponentiation in $\mathbb{G}_T$, exponentiation/multiplication in $Z_{n^{sp+1}}$, bilinear pairing, hash function, and addition/subtraction/multiplication/division/exponentiation in Z_p. We show the implemented running time of each entity listed in Table V.

1) Registration: During registration, the CA spends 54 milliseconds in generating the elliptic curve and then prepares keys for entities. CA chooses $(x_i)_{i\in[\ell]}$, computes $\mathsf{pk}_{\mathcal{R}}$, picks $\acute{q}, (\acute{p}_i)_{i\in[\ell']}$, and computes Q_1 and Q_2 for a supplier S_i. It consists of $(\ell + 2 + \ell')Mu_1$ which takes approximately 0.03 seconds. CA computes ss_j, vk_j, and δ for a platform P_j. It contains $5Exp + Add + (n_1! + 1)Mul$ which takes 0.98 milliseconds.

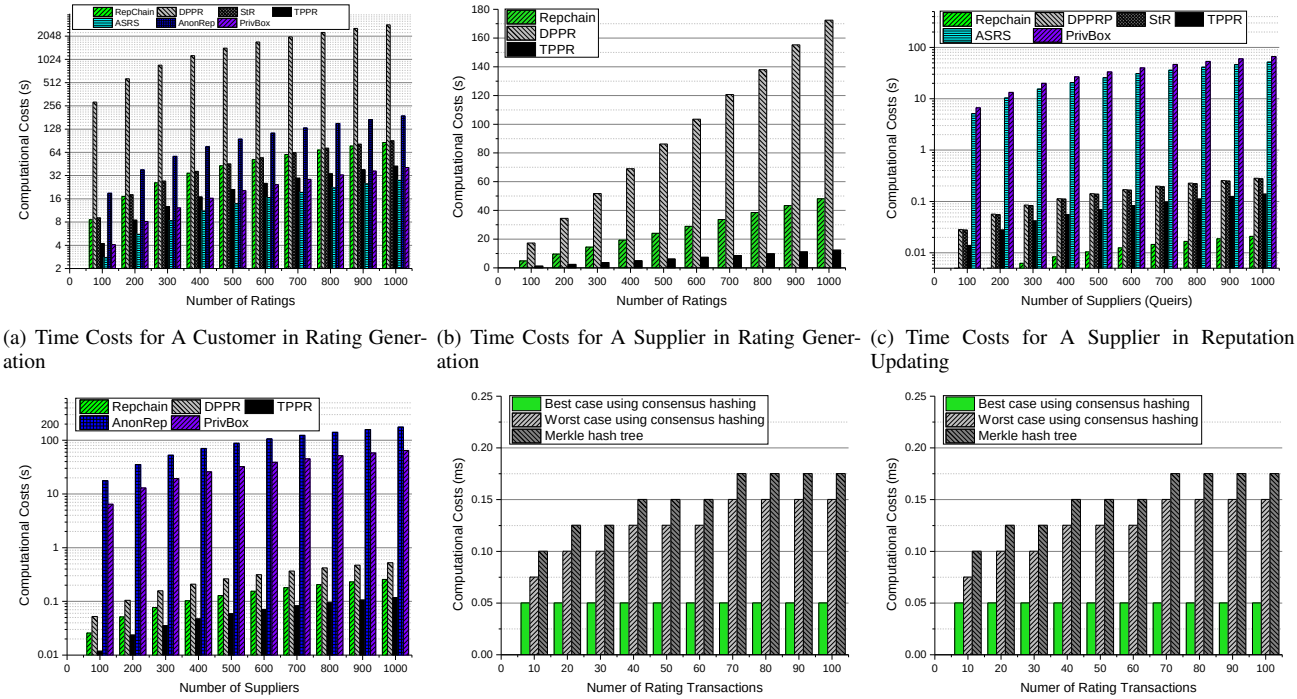
2) Rating Generation: When generating a rating transaction, a customer performs $19Mu_1 + 2Ad_1 + 9Ex + 6Ad + 4Sub + 4Mul + 2H + 5BP + Mu_T + 2Ex_T$ cryptographic operations in rating generation and a supplier performs $6Mu_1 + 2Ad_1 + 3BP + 2Mul_T + 3Ex_T + Div + l(Ad + Mul)$. As shown in Table V, it only requires 86 milliseconds for a customer and 43 milliseconds for a supplier in rating generation. Since generating blind signatures is right after financial transactions and before the blockchain phase, thereby, it does not affect the maintenance of the CBC. We use an example to explain the number of cryptographic operations. A supplier S_j performs $6Mu_1 + 2Ad_1 + 3BP + 2Mul_T + 3Ex_T + Div + l(Ad + Mul)$ in rating generation: 1. Compute X and X_i: $2Mu_1 + Ad + Div$; 2. Verify D^1_{ij}: $2Mu_1 + 2Ad_1$; 3. Verify D^2_{ij}: $3BP + 3Ex_T + 2Mul_T$; and 4. Compute σ_{ji}: $2Mu_1 + Mul + (l-1)(Ad + Mul)$.

We compare our RepChain with the existing schemes, i.e., DPPR [3], StR [29], ASRS [5], TPPR [31], and PrivBox [7]. There are a querying agent and n_2 source agents involved in the rating generation of DPPR, and they act like supplier and customer. A DPPR customer needs to encrypt $n_2 + 1$ shares with his public key and n_2 share with n_2 source agents' public keys using Paillier cryptosystem, compute a product of $n_2 + 1$ encrypted shares, and generate a set membership zero-knowledge proof to prove the correctness of a share. A DPPR supplier has to verify the proof received from $n_2 + 1$ source agents. As shown in Figs. 5(a)-5(b), the computational costs of a customer and a supplier in RepChain are moderate. In Fig. 5(b), Repchain is compared with only two schemes because the other schemes do not have the role supplier in their system model. RepChain outperforms DPPR because a DPPR customer needs to split a rating into n_2+1 shares, which incur extra computations while RepChain does not require this operation. A StR customer only encrypts ratings using Paillier cryptosystem, and it spends 0.1 milliseconds more than a RepChain customer. However, it cannot defend from false rating attack and multiple rating attack given their security model. An ASRS customer has the lowest computational cost for adopting BBS signatures, but it only provides identity anonymity and cannot resist multiple rating attacks. TPPR shows a low computational cost for asking a customer to interact with a supplier to generate a blind signature [32], but it cannot resist a false rating attack. An AnonRep customer has a higher computational cost because a time-consuming ring signature scheme is used [58]. The suppliers in ASRA, AnonRep, and PrivBox do not operate in this phase.

3) Reputation Updating: A supplier conducts $Add+Mul+Div$ in updating reputation and a platform conducts $2Exp + 5Mul$ for one supplier. As shown in Fig. 5(c), it only requires 86 milliseconds for a supplier in reputation updating, and it

TABLE V
RUNNING TIME (UNIT: MILLISECOND)

Phase	Registration		Rating Generation		Updating Reputation		Rating Verification
	Supplier	Platform	Customer	Supplier	Platform	Supplier	Verifier
Runtime	30	0.98	86	43	0.25	0.016	0.05



(a) Time Costs for A Customer in Rating Generation

(b) Time Costs for A Supplier in Rating Generation

(c) Time Costs for A Supplier in Reputation Updating

(d) Time Costs for A Platform in Reputation Updating

(e) Verification Efficiency

(f) Monetary Cost

Fig. 5. Performance Analysis

outperforms other schemes since they involve extra decryption. StR, ASRS, and AnonRep do not include a platform or a similar role in this phase and TPPR nodes are modeled as platforms. As shown in Fig. 5(d), it costs 43 milliseconds for a platform in reputation updating which is less than half of that for a DPPRP customer which shares the same functional purpose as a platform. In PrivBox, a bulletin board, i.e., platform computes the reputation.

4) Rating Verification: A rating verifier performs $(|\{CH_o|For\ o = Height, o \in min\{o'\} \wedge o >= j\}|)Hash$ in verifying a rating transaction i where o' is the next o to be selected. This step costs 0.05 milliseconds and 0.075 milliseconds for verifying one rating using consensus hashing in the best case and worst case, respectively. The result of comparison with the Merkle hash tree method is recorded in Fig. 5(e).

C. Communication Overhead

We analyze the communication overhead of customer, supplier, platform, and elected platform by counting the number of kbytes. We show the communication overhead of each entity and comparison with existing work in Table VI.

During rating generation, a customer C_i sends two proofs pf_{ij}^1, pf_{ij}^2 and a rating transaction $Tx_{ij} = (S_j, ct_{ij}, rc_i)$, which have a total binary length of 1.081 kbytes. A supplier S_j sends $X, \{X_i\}$ and a signature $\sigma_{ji} = (A, B, \hat{B})$ to C_i, which have a transmission length of 0.732 kbytes. In updating reputation, a platform P_k sends sd_{kj} and a signature σ_{kj} for supplier S_j. The total binary length is 0.25 kbytes. In maintaining the blockchain, an elected platform sends $0.25 * n_2 + 0.277$ kbytes by sending share combining to n_2 suppliers and packing a new block.

DPPR and StR suffer from a high communication overhead for customers and suppliers because they both adopted a sharing mechanism that incurs heavy communication overhead. Such a mechanism requires a supplier to send a set of identities of source agents, i.e., customers, to all other source agents. All the source agents have to compute at least n_1 encryptions of their shares. TPPR also has a high communication overhead for customers for using blind signatures, although suppliers do not send too much data. In ASRS and AnonRep, only customers send a plaintext rating and a group signature to the server. A PrivBox customer has to send two identities, an encrypted rating, a token, and a 1-out-2 non-interactive zero-knowledge (NIZK) proof. A PrivBox supplier only sends reputation access queries to platform, which is excluded in comparison.

TABLE VI
COMMUNICATION OVERHEAD OF ENTITIES (UNIT: KBYTES)

Scheme	C	S	P	$\hat{P}$
RepChain	1.081	0.732	$0.375 * n_2$	$0.25 * n_2 + 0.277$
DPPR [3]	26.162	22.141	n/a	n/a
StR [29]	12.5	8.545	n/a	n/a
ASRS [5]	1.5	n/a	n/a	n/a
AnonRep [6]	5.12	n/a	n/a	n/a
TPPR [31]	16.532	0.168	$1.081 * n_2$	$0.289 * n_2$
PrivBox [7]	1	n/a	n/a	n/a

D. Monetary Cost

The monetary cost comes from the gas cost of operating smart contracts. The gas price in our blockchain is $1 * 10^{-9}$ Ether and the Ether price on February 11, 2021 is \$1734 according to coinmarketcap real-time table (https://coinmarketcap.com/). The gas cost of each transaction is shown in Table VII. For example, the Request transaction costs $2.73{*}10^{-4}$ Ether, which is equal to 0.47 USD.

TABLE VII
MONETARY COST

Function	Request	Collect	Update	Complete	Reject
Gas	272524.8	169294.4	40618	37474	23767
Ether($*10^{-4}$)	2.73	1.69	4.06	3.74	0.24
USD	0.47	0.29	0.07	0.06	0.04

VIII. DISCUSSIONS

We provide some discussions in this section to what we could further improve the RepChain in future work.

A. Textual Comments

In this work, we primarily focus on providing an aggregate numerical rating towards suppliers. In some cases, customers need more than ratings to express their good/bad feelings and there is not always a direct correlation between reputation and the ratings. Therefore, we plan to allow customers to leave textual comments and attach some semantic context to the ratings. In this way, the whole reputation system will become more powerful and compatible with existing reputation systems. However, doing so will incur new privacy challenges since textual comments may contain sensitive information about customers. In addition, simple encryption of comments cannot eradicate privacy concerns. These issues hinder the wide application of textual comment-supported systems. To overcome these limitations, we will improve RepChain to support numerical ratings and textual comments in a privacy-preserving way.

B. Advantages of Consortium Blockchain

We use a consortium blockchain to keep track of all the rating transactions. Such a blockchain has several advantages over a joint centralized database: (1) Transparency. Users' reputations are publicly updated in a distributed manner. This decentralized model defends against the single point of failure/attack and prevents the centralized database from falsifying users' reputations. (2) Usability. Users can easily look up and verify the reputation of a user from a different platform which is encapsulated in the platform's database before.

C. Trust Among Consortium Members

Blockchain is a solution wherever and whenever there are commercial opportunities produced by the collaboration between mutually distrusting parties. Bitcoin is the first practical blockchain application that enables electronic transactions among individuals who do not fully trust each other. While public blockchains establish trust among individuals, consortium blockchains build trust between enterprises. It increases the profitability for each enterprise by relying on technically sound techniques. If some consortium member recruits some raters to issue false ratings, we can mitigate this attack by adopting an enrollment fee where the adversary joins the system [36]. How to choose the appropriate fee amount and maintain incentive compatibility is left as an open research challenge. It is not in its best interest to improve its reputation in the long run. Furthermore, any consortium member cannot issue rating tokens for transactions since we have used one-time rating credentials.

D. Implementation Extension

In this work, we use Ethereum smart contracts to implement various functionalities of our proposed system. It is important to consider extending them to other platforms such as Hyperledger, RSK Smart Bitcoin (RSK RBTC), and Entrepreneurial Operating System (EOS). Fortunately, some platforms already support the extension. For example, Hyperledger Fabric now supports Ethereum virtual machine bytecode smart contracts [59]. Its contracts can be written in Solidity and Fabric has a corresponding web3 provider for developing decentralized applications using web3.js. For the other platforms which do not have such support, it remains a task to rewrite the same smart contract logic in a different programming language that is supported by the target blockchain platform. Moreover, the automatic translation of smart contracts between different blockchain platforms is also an interesting research topic.

E. Consortium Mechanism

We choose Ethereum to be the blockchain platform in experiments for its wide adoption. The kind of blockchain is consortium blockchain and we use PoS in the design. In a real deployment, we do not rely on a specific consensus mechanism since there are several mature mechanisms in the literature. In other words, we can choose PoW, PoS, PBFT, and other mechanisms to complete the consensus.

IX. CONCLUSIONS AND FUTURE WORK

In this work, we have proposed RepChain: a blockchain-based privacy-preserving reputation system for E-commerce platforms. RepChain realizes collaborations among different e-commerce platforms. Specifically, we utilize one-show anonymous credentials constructed from two-move blind signatures,

zero-knowledge range proofs, secure multiparty computation, blockchain, smart contract, and consensus hashes. The security and privacy analysis validates that RepChain protects rating privacy, identity privacy, and unlinkability. It also resists to multiple rating attack and abnormal rating attacks. The experimental results show that the computational costs and communication overhead of RepChain are moderate compared to existing work. It only costs users a small amount of monetary costs when interacting with the blockchain.

For future work, we will consider building an anonymous and verifiable reputation system supporting additional commenting. Additional commenting refers to the scenario where a rater needs to comment on a previously rated good. In this case, we have to guarantee that the second comment can not be linked to the previous rating. In the experiment, we realized a prototype of RepChain and showed its feasibility. But still, it remains a future work to enhance RepChain's practicality by integrating it with real-world services and their real e-commerce web interfaces (e.g., Amazon and ebay).

ACKNOWLEDGMENT

The work described in this paper was supported by National Natural Science Foundation of China (NSFC) under the grant No. 62002094, and Anhui Provincial Natural Science Foundation under the grant No. 2008085MF196. It is partially supported by EU LOCARD Project under Grant H2020-SU-SEC-2018-832735. This work was carried out during the tenure of an ERCIM 'Alain Bensoussan' Fellowship Programme granted to Dr. Meng Li.

REFERENCES

[1] "eBay Uses 100% Open Source WSO2 ESB to Process More Than 1 Billion Transactions Per Day." Available: https://wso2.com/casestudies /ebay-uses-100-open-source-wso2-esb-to-process-more-than-1-billion-transactions-per-day/.

[2] "Chinese bike-sharing start-up ofo says it's now worth more than $2 265 billion." 2017. https://www.cnbc.com/2017/04/17/ofo-chinese-bike-sharing-start-up-says-its-now-worth-more-than-2-billion.html.

[3] O. Hasan, L. Brunie, E. Bertino, and N. Shang, "A decentralized privacy preserving reputation protocol for the malicious adversarial model," *IEEE Trans. Information Forensics and Security (TIFS)*, 2013, 8 (6): 949-962.

[4] P. Resnick and R. Zeckhauser, "Trust among strangers in internet transactions: Empirical analysis of ebay' s reputation system." *The Economics of the Internet and E-Commerce*, 2002, 11 (2): 127-157.

[5] J. Blömer, J. Juhnke, and C. Kolb, "Anonymous and publicly linkable reputation systems," *Proc. 19th International Conference on Financial Cryptography and Data Security (FC)*, January 2015: 478-488, San Juan, Puerto Rico.

[6] E. Zhai, D. I. Wolinsky, R. Chen, E. Syta, C. Teng, and B. Ford, "AnonRep: Towards Tracking-Resistant Anonymous Reputation," *Proc. 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2016: 583-596, Santa Clara, USA.

[7] M. A. Azad, S. Bag, and F. Hao, "PrivBox: Verifiable decentralized reputation system for online marketplaces," *Future Generation Computer Systems*, 2018, 89: 44-57.

[8] "Why credit scores differ between credit-reporting agencies." 2018. https://www.creditkarma.com/advice/i/why-credit-scores-differ-between-credit-reporting-agencies.

[9] C. Y and D. Zhu, "Fraud detections for online businesses: A perspective from blockchain technology," *Financial Innovation*, 2016, 2 (1): 1-20.

[10] "China's Didi Chuxing suspends carpool service after woman is killed." 2018. http://fortune.com/2018/08/26/didi-chuxing-carpool-suspended-woman-killed/.

[11] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2009. https://bitcoin.org/bitcoin.pdf.

[12] W. Hao, J. Zeng, X. Dai, J. Xiao, Q.-S. Hua, H. Chen, K.-C. Li, and H. Jin, "Towards a trust-enhanced blockchain P2P topology for enabling fast and reliable broadcast," *IEEE Trans. Network and Service Management (TNSM)*, 2020, 17 (2): 904-917.

[13] S. K. Singh, S. Rathore, and J. H. Park, "BlockIoTIntelligence: A blockchain-enabled intelligent IoT architecture with artificial intelligence," *Future Generation Computer Systems*, 2020, 110: 721-743.

[14] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Blockchain for 5G and beyond networks: A state of the art survey," *Journal of Network and Computer Applications*, 2020, 166: 102693.

[15] H. R. Hasan and K. Salah, "Blockchain-based solution for proof of delivery of physical assets," *Proc. 1st International Conference on Blockchain (ICBC)*, July/August 2018: 139-152, Seattle, USA.

[16] K. Salah, N. Nizamuddin, R. Jayaraman, and M. Omar, "Blockchain-based soybean traceability in agricultural supply chain," *IEEE Access*, 2019, 7: 73295-73305.

[17] A. Kumar, C. Fischer, S. Tople, and P. Saxenam, "Traceability analysis of Monero' blockchain," *Proc. 22nd European Symposium on Research in Computer Security (ESORICS)*, September 2017: 153-173, Oslo, Norway.

[18] M. Li, L. Zhu, and X. Lin, "CoRide: A privacy-preserving collaborative-ride hailing service using blockchain-assisted vehicular fog computing," *Proc. ACM 15th EAI International Conference on Security and Privacy in Communication Networks (SecureComm)*, October 2019: 408-422, Orlando, USA.

[19] M. Li, C. Lal, M. Conti, and D. Hu, "LEChain A Blockchain-based Lawful Evidence Management Scheme for Digital Forensics," *Future Generation Computer Systems*, September 2020. DOI: 10.1016/j.future.2020.09.038.

[20] DREP - A Blockchain-Based Decentralized Reputation System." Available: https://raw.githubusercontent.com/drep-project/one-paper/master /DREP%20WhitePaper_EN.pdf.

[21] "Internet privacy concerns affecting online shopping, banking habits," May 20, 2016. Avaliable: https://www.goldenfrog.com/blog/internet-privacy-concerns-affecting-online-shopping-banking-habits-us.

[22] G. Fuchsbauer, C. Hanser, and D. Slamanig. "Practical round-optimal blind signatures in the standard model," *Proc. 35th International Cryptology Conference (CRYPTO)*, August 2015: 233-253, Santa Barbara, USA.

[23] J. Camenisch, R. Chaabouni, and a. shelat. "Efficient protocols for set membership and range proofs," *Proc. 14th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT)*, December 2008: 234-252, Melbourne, Australia.

[24] C. Miao, W. Jiang, L. Su, Y. Li, S. Guo, Z. Qin, H. Xiao, J. Gao, and K. Ren, "Cloud-enabled privacy-preserving truth discovery in crowd sensing systems," *Proc. 13th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, November 2015: 183-196.

[25] M. Andrychowicz, S. Dziembowski, D. Malinowski, and L. Mazurek, "Secure multiparty computations on bitcoin," *Proc. 35th IEEE Symposium on Security and Privacy (S&P)*, May 2014: 443-458, San Jose, USA.

[26] I. Damgård and M. Jurik, "A generalisation, a simplication and some applications of paillier's probabilistic public-key system," *International Workshop on Practice and Theory in Public Key Cryptography: Public Key Cryptography, Springer-Verlag*, 2001: 119-136.

[27] M. Ali, J. Nelson, R. Shea, and M. J. Freedman, "Blockstack: A global naming and storage system secured by blockchains," *USENIX Annual Technical Conference (USENIX ATC)*, 2016: 181-194.

[28] "Ethereum." 2018. https://www.ethereum.org.

[29] T. Dimitriou and A. Michalas, "Multi-party trust computation in decentralized environments in the presence of malicious adversaries," *Ad Hoc Networks*, 2014, 15: 53-66.

[30] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes" *J. Stern (Ed.), EUROCRYPT, Lecture Notes in Computer Science*, 1999: 223-238.

[31] A. Schaub, R. Bazin, O. Hasan, and L. Brunie, "A trustless privacy-preserving reputation system," *International Information Security&Privacy*, 2016: 398-411.

[32] D. Chaum, A. Fiat, and M. Naor, "Untraceable electronic cash," *Proc. International Cryptology Conference* , 1988: 319-327.

[33] R. Dennis and G. Owen, "Rep on the block: A next generation reputation system based on the blockchain," *Proc. 11th Internet Technology & Secured Transactions*, 2016: 131-138, Barcelona, Spain.

[34] F. Casino and C. Patsakis, "An efficient blockchain-based privacy-preserving collaborative filtering architecture," *IEEE Trans. Engineering Management*, 2020, 67 (4): 1501-1513.

[35] B. D. A. Kiayias, A. Russell and R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol," *Proc. 37th International Cryptology Conference*, August 2017: 357-388, Santa Barbara, USA. (CRYPTO).

[36] A. Kosba, A.Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," *Proc. IEEE Symposium on Security and Privacy (S&P)*, May 2016, pp. 839-858.

[37] L. Zhu, M. Li, Z. Zhang, and Z. Qin, "ASAP: An anonymous smart-parking and payment scheme in vehicular networks," *IEEE Trans. Dependable and Secure Computing (TDSC)*, July 2020, 17 (4): 703-715. DOI: 10.1109/TDSC.2018.2850780.

[38] M. Li, L. Zhu, and X. Lin, "Privacy-preserving traffic monitoring with false report filtering via fog-assisted vehicular crowdsensing," *IEEE Trans. Services Computing (TSC)*, March 2019, PP (99): 1-11. DOI: 10.1109/TSC.2019.2903060.

[39] M. Li, Y. Chen, S. Zheng, D. Hu, C. Lal, and M. Conti, "Privacy-preserving navigation supporting similar queries in vehicular networks," *IEEE Trans. Dependable and Secure Computing (TDSC)*, 2020, PP (99): 1-16. DOI: 10.1109/TDSC.2020.3017534.

[40] Y. Chen, M. Li, S. Zheng, D. Hu, C. Lai, and M Conti, "One-time, oblivious, and unlinkable query processing over encrypted data on cloud" *Proc. 22nd International Conference on Information and Communications Security (ICICS)*, August 2020: 350-365, Copenhagen, Denmark.

[41] M. Li, J. Gao, Y. Chen, J. Zhao, and M. Alazab, "Privacy-preserving ride-hailing with verifiable order-linking in vehicular networks," *Proc. 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, December 2020: 599-606, Guangzhou, China.

[42] M. Li, L. Zhu, and X. Lin, "Efficient and privacy-preserving carpooling using blockchain-assisted vehicular fog computing," *IEEE Internet of Things Journal (IoTJ)*, 2019, 6 (3): 4573-4584. DOI: 10.1109/JIOT.2018.2868076.

[43] L. Zhu, M. Li, and Z. Zhang, "Secure fog-assisted crowdsensing with collusion resistance: From data reporting to data requesting," *IEEE Internet of Things Journal (IoTJ)*, March 2019, 6 (3): 5473-5484. DOI: 10.1109/JIOT.2019.2902459.

[44] E. Androulaki et al., "Hyperledger fabric: A distributed operating system for permissioned blockchains," *Proc. 13th European Conference on Computer Systems (EuroSys)*, April 2018: 1-15.

[45] Z. Yang, K. Yang, L. Lei, K. Zheng, and V.. M. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet of Things Journal*, 2018, 6 (2): 1495-1505.

[46] J. Kang , R. Yu, X. Huang, S. Maharjan, Y. Zhang, and E. Hossain, "Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains," *IEEE Trans. Industrial Informatics*, 2017, 13 (6): 3154-3164.

[47] F. Baldimtsi and A. Lysyanskaya, "Anonymous credentials light," *Proc. 20th ACM Conference on Computer and Communications Security (CCS)*, November 2013: 1087-1098, Berlin, Germany.

[48] T. H. Yuen, Q. Huang, Y. Mu, W. Susilo, D. Wong, and G. Yang, "Efficient non-interactive range proof," *Proc. International Computing and Combinatorics Conference*, 2009: 138-147.

[49] K. Peng and F. Bao"Batch range proof for practical small ranges," *Proc. International Conference on Cryptology in Africa (AFRICACRYPT)*, May 2010: 114-130, Stellenbosch, South Africa.

[50] P. Tsankov, A. Dan, D. Drachsler-Cohen, A. Gervais, F. Bönzli, and M. Vechev, "Securify: Practical security analysis of smart contracts," *Proc. 25th ACM SIGSAC Conference on Computer and Communications Security (CCS)*, October 2018: 67-82, Toronto, Canada.

[51] R. Almadhoun, M. Kadadha, M. Alhemeiri, M. Alshehhi and K. Salah, "A user authentication scheme of IoT devices using blockchain-enabled fog nodes," *Proc. IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, October/November 2018: 1-8, Aqaba, Jordan.

[52] H. R. Hasan and K. Salah, "Proof of delivery of digital assets using blockchain and smart contracts," *IEEE Access*, 2018, 6: 65439-65448.

[53] A. Permenev, D. Dimitrov, P. Tsankov, D. Drachsler-Cohen, and M. Vechev, "VerX: Safety verification of smart contracts," *Proc. 41st IEEE Symposium on Security and Privacy (S&P)*, 2020: 1661-1677, San Francisco, USA.

[54] VERISMART: A highly precise safety verifier for Ethereum smart contracts," *Proc. 41st IEEE Symposium on Security and Privacy (S&P)*, 2020: 1678-1694, San Francisco, USA.

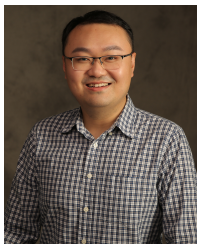
[55] "Ethereum Wallet and Mist Beta 0.11.1 - windows hotfix." https://github.com/ethereum/mist/releases.

[56] "Download Geth No Nick (v 1.8.19)." https://ethereum.github.io/go-ethereum/downloads.

[57] D. M. Kar and I. Ray, "Systematization of knowledge and implementation: Short identity-based signatures, *https://arxiv.org/abs/1908.05366?context=cs*, 2019: 1-21.

[58] J. K. Liu and D. S. Wong, "Linkable ring signatures: Security models and new schemes," *Proc. Computational Science and Its Applications (ICCSA)*, May 2005: 614-623.

[59] Hyperledger, "Hyperledger Fabric Now Supports Ethereum." 2018. Available: https://www.hyperledger.org/blog/2018/10/26/hyperledger-fabric-now-supports-ethereum.



Meng Li received the B.E. degree in the Information Security from Hefei University of Technology in 2010, received the M.S. degree from Computer Science and Technology from Beijing Institute of Technology in 2013, and received the Ph.D. degree in Computer Science and Technology from Beijing Institute of Technology in 2019. He is now an Associate Professor in the School of Computer Science and Information Engineering, Hefei University of Technology, China. He is also a postdoctoral fellow at Department of Mathematics, University of Padua, Italy, where he is with the SPRITZ research group. He was sponsored by the ERCIM 'Alain Bensoussan' Fellowship Programme in October 2019 to conduct postdoctoral research at CNR, Italy. He was sponsored by the China Scholarship Council (CSC) to study in the Broadband Communications Research (BBCR) Lab at University of Waterloo and Wilfrid Laurier University from September 2017 to August 2018. His research interests include security and privacy, fairness, vehicular networks, applied cryptography, edge computing, and blockchain. In this area, he published more than 30 papers in international peer-reviewed journals and conference, including TDSC, TSC, TII, IoT Journal, Information Sciences, IEEE Communications Magazine, IEEE Wireless Communications, MobiCom, ICICS, SecureComm, TrustCom, and IPCCC.

Liehuang Zhu is a full professor in the School of Cyberspace Science and Technology, Beijing Institute of Technology. He is selected into the Program for New Century Excellent Talents in University from Ministry of Education, P.R. China. His research interests include Internet of Things, Cloud Computing Security, Internet and Mobile Security. He has published over 60 SCI-indexed research papers in these areas, as well as a book published by Springer. He serves on the editorial boards of three international journals, including IEEE Internet-of-Things Journal, IEEE Network, and IEEE Transactions on Vehicular Technology. He won the Best Paper Award at IEEE/ACM IWQoS 2017 and IEEE TrustCom 2018.

Zijian Zhang (zhangzijian@bit.edu.cn) received the Ph.D. degree from the School of Computer Science and Technology at the Beijing Institute of Technology. He is now a research fellow with the School of Computer Science at the University of Auckland. He was a visiting scholar in the Computer Science and Engineering Department of the State University of New York at Buffalo in 2015. His research interests include design of authentication and key agreement protocol and analysis of entity behavior and preference.

Chhagan Lal is currently working as a postdoctoral research fellow at Department of Intelligent Systems, CyberSecurity Group, TU Delft, Netherlands Previously, he was a postdoctoral research fellow at Simula Research Laboratory, Norway. He was a postdoctoral fellow at Department of Mathematics, University of Padua, Italy, where he was part of the SPRITZ research group. He received his PhD in Computer Science and Engineering from the Malaviya National Institute of Technology, Jaipur, India, in 2014. During his PhD, he has been awarded with the Canadian Commonwealth Scholarship under the Canadian Commonwealth Scholarship Program to work in University of Saskatchewan, Saskatoon, SK, Canada. His current research areas include applications of blockchain technologies, security in software-defined networking, and Internet of Things networks.

Mauro Conti is Full Professor at the University of Padua, Italy. He is also affiliated with TU Delft and University of Washington, Seattle. He obtained his Ph.D. from Sapienza University of Rome, Italy, in 2009. After his Ph.D., he was a Post-Doc Researcher at Vrije Universiteit Amsterdam, The Netherlands. In 2011 he joined as Assistant Professor the University of Padua, where he became Associate Professor in 2015, and Full Professor in 2018. He has been Visiting Researcher at GMU, UCLA, UCI, TU Darmstadt, UF, and FIU. He has been awarded with a Marie Curie Fellowship (2012) by the European Commission, and with a Fellowship by the German DAAD (2013). His research is also funded by companies, including Cisco, Intel, and Huawei. His main research interest is in the area of Security and Privacy. In this area, he published more than 350 papers in topmost international peer-reviewed journals and conferences. He is Area Editor-in-Chief for IEEE Communications Surveys & Tutorials, and Associate Editor for several journals, including IEEE Communications Surveys & Tutorials, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Information Forensics and Security, and IEEE Transactions on Network and Service Management. He was Program Chair for TRUST 2015, ICISS 2016, WiSec 2017, ACNS 2020, and General Chair for SecureComm 2012, SACMAT 2013, CANS 2021, and ACNS 2022. He is Senior Member of the IEEE and ACM. He is member of the Blockchain Expert Panel of the Italian Government. He is Fellow of the Young Academy of Europe.

Mamoun Alazab is an Associate Professor at the College of Engineering, IT and Environment at Charles Darwin University, Australia. He received his PhD degree in Computer Science from the Federation University of Australia, School of Science, Information Technology and Engineering. He is a cybersecurity researcher and practitioner with industry and academic experience. Alazab's research is multidisciplinary that focuses on cybersecurity and digital forensics of computer systems with a focus on cybercrime detection and prevention including cyber terrorism and cyber warfare. He has more than 150 research papers. He delivered many invited and keynote speeches, 24 events in 2019 alone. He convened and chaired more than 50 conferences and workshops. He works closely with government and industry on many projects, including Northern Territory (NT) Department of Information and Corporate Services, IBM, Trend Micro, the Australian Federal Police (AFP), the Australian Communications and Media Authority (ACMA), Westpac, United Nations Office on Drugs and Crime (UNODC), and the Attorney Generals Department. He is a Senior Member of the IEEE. He is the founding chair of the IEEE Northern Territory (NT) Subsection.