A Secure Personal-Data Trading System Based on Blockchain, Trust, and Reputation

***Abstract*—Every citizen has the right to privacy and, therefore, the right to control their personal information without the interference of organizations. They can inform and trade their data, deciding to whom, when, and where their information is available. The current solutions that use centralized trust in trading systems, however, restrict user's control over their proprietary data. This paper proposes a secure, agile, and effective system to a distributed, automatic, and transparent data trading between domains using blockchain, smart contracts, trust, and reputation. We develop and implement a prototype of a trust and reputation system based on real-life interactions. The results show that the proposed system provides security and privacy to protect data trading between domains in a quick and distributed way, performing at hundreds of transactions per second, and effectively punishing malicious behavior.**

I. Introduction

Personal-data access control is becoming a major concern as devices increasingly delegate the storage of sensitive data to cloud-based centralized authorities [1]. The centralized authorities, such as governments and companies, may then access, control, and share personal user data with third parties. For the owner of the data, using centralized data-storage solutions implies losing control of personal data, paying high fees, signing terms that often compromise privacy, and still be subject to data leaks. Besides, even if the centralized authority behaves honestly, malicious users often compromise cloud-based services through internal attacks and denial of service (DoS) attacks. A more efficient way to ensure security and privacy while preserving the owner's control over the data is to use blockchain technology, which provides a distributed and auditable solution for storing personal data.

In a previous article [2], we proposed a blockchain-based system to commercialize data in a safe, automatic, and distributed way. The system uses the integrity and auditability properties of the blockchain to store each users' access permissions to private and sensitive data. The scenario considers large data centers interconnected through software-defined network technology (SDN), which implements data access control.

Despite maintaining the record immutability of assets transfer, the blockchain does not guarantee the delivery of data stored off-chain nor their quality. Malicious sellers can take advantage of this vulnerability to advertise false data or fail to deliver data acquired by honest buyers. Thus, the system is unable to prevent malicious behavior by sellers, causing damages to buyers who take legitimate actions. A trust and reputation system (TRS) allows the identification of malicious behaviors, in addition to presenting an insight into the quality of the data from buyers' evaluations. A reputation system integrated with the blockchain can provide assessments of data sellers' reputation as well as the quality of the data advertised in a transparent and distributed manner. In this way, buyers can quickly check the reputation of an advertisement and decide whether or not to acquire the data.

This paper proposes a secure data marketplace system based on blockchain, reputation, and trust. The main contributions are as follows:

- The creation of a data marketplace environment in an automatic and distributed way between domains, in which data owners and buyers can advertise and acquire data through the blockchain. The blockchain immutably records transfers, allowing the owner to maintain control over who has access to their data. The proposed marketplace scheme is automatic through smart contracts.
- A reputation and trust system that considers the history of interactions and the participants' opinions to build a seller's reputation. The system adapts the model to the data marketplace scenario, introducing a reputation punishment proportional to the price of the advertised product.
- An efficient data trading model for advertising, purchasing, and evaluating data quality based on transactions in the blockchain. The transactions include the assessment of the seller and the quality of the advertised data.
- Development of the transaction model and trust and reputation models. Performance evaluation results of a developed prototype show that the proposed system is effective in punishing the malicious behavior of a seller.

We organize the rest as follows. Section II discusses related work. Section III details the attacker model for the entities involved in the data trading system. Section IV presents the proposed models of trust and reputation and the integration of these models with the blockchain. Section V describes the required procedures to integrate trust and reputation with blockchain. Section VI evaluates the performance of the system proposed by simulations and discusses the results. Finally, Section VII concludes the article and provides directions for future work.

II. Related Work

Several works investigate the use of trust and reputation to provide secutiy in computer networks. Velloso *et al.* propose

TABLE I
ATTACKS AND COUNTERMEASURES PROPOSED FOR THE REPUTATION SYSTEM

Attacks	Description	Countermeasures
Bad-mouthing attack	Malicious users publish false reports about a target user to affect the reputation of the target. This attack is most powerful when there is collusion between malicious nodes.	To use a dissatisfaction flag by a seller dissatisfied with an evaluation. The system logs requests and detects malicious nodes.
On-off attack	A malicious user changes his/her behavior from legitimate to malicious and vice versa to damage the network without being identified.	To use an adaptive forgetting factor in reputation calculation.
Sybil attack	An attacker generates multiple false identities to increase his influence in the system. An attacker can use these identities to perform a bad-mouthing attack on a target user.	To use a permissioned blockchain and enforcing regulations to sellers on a per-organization basis.
Newcomer attack	An attacker generates new identities repeatedly and pretends to be a new user, to continue acting maliciously without being punished.	To use a permissioned blockchain and enforcing regulations to sellers on a per-organization basis.
Conflicting-behavior attack	An attacker can behave differently according to each neighbor, creating conflicting recommendations from well-behaving nodes.	All users' feedbacks are publicly verifiable as transactions in the blockchain.

a trust model based on human interactions to establish trust among nodes in a ad hoc network [3]. Kamvar *et al.* propose the Eigentrust algorithm that assigns a global trust value to each peer in a peer-to-peer (P2P) file-sharing system. The algorithm is based on the past file uploads of each peer and uses indirect trust to calculate the global trust view of the system. Sun *et al.* presents defense mechanisms against attacks on trust and reputations in a mobile ad hoc network (MANET) [4].

The blockchain technology can provide auditability and traceability in trustless environments [5], [6]. We consider the concepts of the cited works and adapt them to a create a trust and reputation system that considers the properties of blockchains and distrubuted trustless environments.

Other works apply trust and reputation to blockchain. Oliveira *et al.* propose a blockchain reputation-based consensus (BRBC) [7]. Dennis and Owen propose a file transfer reputation system in which the blockchain publicly stores users' recommendations [8]. The authors, however, do not offer a solution to the on-off attack and do not implement their proposal. Malik *et al.* propose a framework to manage participants' trust in a blockchain that records transfers in a supply chain [9]. Buyers, government authorities, and sensors attest to the quality of the product by issuing evaluations through transactions in the blockchain. A smart contract calculates a trader's reputation using a weighted sum of the entities' ratings. The authors limit their proposal to the case of a supply chain and disregard the product price when calculating reputation. Furthermore, the proposal is susceptible to on-off attacks, in which a user changes his behavior between legitimate and malicious, damaging the network without being detected.

Putra *et al.* propose a trust and reputation management system for blockchain-based systems to control the access to IoT devices [10]. The system uses smart contracts to assess the trust and reputation of each node, detecting and eliminating malicious nodes from the network. The authors use reputation as an attribute to control access to a device, defining a minimum reputation that a participant must have to guarantee access to data. However, the proposal is susceptible to on-off attacks and does not reward the data owners. Moreover, the commercialization of data by the owners is a desirable property.

Unlike the cited articles, this paper proposes a trust and reputation system based on blockchain that is effective in punishing malicious behavior and adapted to blockchain-based data marketplace. The marketplace system as a whole is secure, agile, and automatic, rewarding the sellers who sell their private data and punishing malicious sellers.

III. ATTACKER MODEL

Attacks on the system of trust and reputation represent an attempt by a malicious node to increase its reputation or damage legitimate nodes reputation. We consider five attacks: (i) bad-mouthing attack; (ii) on-off attack; (iii) Sybil attack; (iv) newcomer attack; and (v) conflicting-behavior attack. Table I presents each attack and the proposed countermeasures.

We consider that a blockchain attacker aims to prevent a participant from adding a legitimate transaction or block to the blockchain. The fault-tolerance property of the consensus protocol requires that the attacker control the majority of organizations to effectively affect the consensus protocol, mitigating this type of attack. The immutability and distribution property of the blockchain structure allow transaction and block issuers to check if their proposal was correctly added to the blockchain.

Attacks on sellers or buyers try to obtain private and sensible advertised data or to impersonate the target. We consider that all advertised data in the blockchain is encrypted, i.e., if the attacker gains access to the data, he/she must acquire the key that decrypts it to obtain personal information about the seller. An attacker may try to impersonate the target to deceive other participants. This attack, however, is not effective once the system requires that all issuers sign their transactions. Furthermore, the blockchain logs every attempt to modify its structure using a stolen pair of keys, allowing the victim to prevent further damage by replacing his/her stolen pair of keys.

Our proposal considers network attacks as an attacker trying to isolate a participant to prevent him/her from issuing

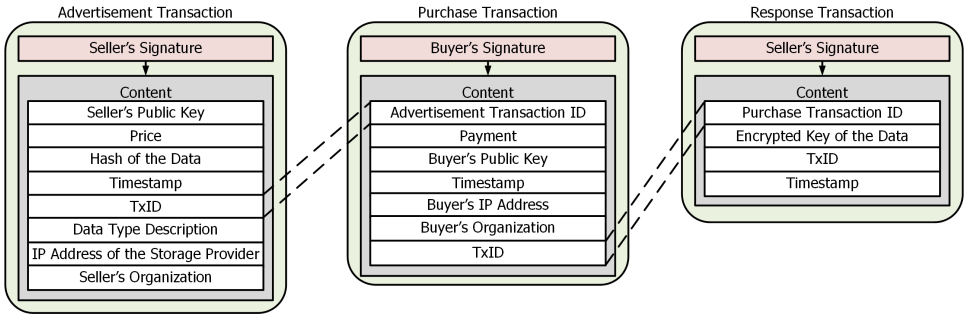


Fig. 1. Our proposed transactions to secure data trading between organizations. Buyers link their purchase transaction with the advertisement using the transaction identifier. After purchasing encrypted data, buyers read the response transaction to acquire the decryption key.

transactions. We mitigate this type of attack by establishing redundant paths between the participants of the blockchain network.

IV. The Proposed Trust and Reputation System

We define trust and reputation as distinct concepts in a distributed system[1]. Trust is a buyer's subjective view of a seller based on their previous interactions. Therefore, each buyer independently calculates and updates his trust in a seller. Conversely, reputation represents a global view of the system concerning a specific seller and consists of aggregating individual trusts of all buyers in the system.

We further separate reputation and trust into seller trust/reputation and data trust/reputation to regulate reputations in the system in a fine-grained and fair manner. For example, decoupling the reputations avoids cases in which a good seller sells low-quality data without being punished, or in which a malicious seller becomes unable to recover his reputation despite selling high-quality data. We define the total reputation of an advertisement as a weighted sum between the reputation of the seller and and the reputation of the data:

$$Rep_{ad} = \alpha Rep_s + (1-\alpha) Rep_{dt}, \tag{1}$$

where α is an adjustable parameter agreed upon by the participants at network startup.

A. Seller Trust

The main idea of our trust model is to simulate real-life social interactions, in which trust gradually increases with positive experiences and decreases significantly whenever a negative experience occurs. For seller trust, we also consider that recent interactions are more relevant than past interactions to allow for possible changes in a seller's behavior. The model uses an adaptive ageing function to implement the gradual forgetfulness of past interactions:

$$I_n = \sum_{i=1}^{n} \beta^{(n-i)} \delta_i, \tag{2}$$

where $0 \leq \beta \leq 1$ is the forgetting factor, n is the total number of interactions that occurred and δ_i is the value associated with each interaction. If the i-th interaction is positive, $\delta_i = \delta_+ > 0$. Otherwise, $\delta_i = \delta_- < 0$. By ensuring $|\delta_-| \gg |\delta_+|$, negative interactions weigh more than positive interactions and we are able to simulate real-life trust.

Unlike previous works that use fixed forgetting factors [9], [10], our work adopts a forgetting factor that adapts according to the probability that the seller will act honestly [4]. We model the probability that a seller will be honest with a beta distribution of prior probability $\frac{1}{2}$, which corresponds to the default probability assumed by the system that an unknown seller will be honest. With each new interaction, the model updates the beta distribution via Bayesian inference and uses the expected value $E[p] = \frac{\delta_+ + 1}{\delta_+ + \delta_- + 2} = \beta$ of the new distribution to estimate the likelihood that the seller will be honest at that time. The main advantage of using an adaptive forgetting factor is preventing on-off attacks, in which a malicious seller behaves well just enough to regain his reputation and behave maliciously once again. With the adaptive forgetting factor, if the probability that the seller is honest is high, i.e. $E[p] \to 1$, the system takes longer to forget the seller's past, rewarding him for his good deeds. If the seller acts maliciously, i.e. $E[p] \to 0$, the system quickly forgets his past actions and the weight of his recent malicious actions on reputation is larger.

After we calculate I_n, we feed it to a Gompertz function to model the trust growth of a buyer i in a seller j:

$$s_{ij} = a \exp(-b \exp(-cI_n)), \tag{3}$$

where a, b, and c are constants that represent the asymptote, the displacement parameter along the x-axis, and the trust growth rate, respectively. We choose the Gompertz function because it increases gradually, simulating real-life trust [10], and we can easily shape it by adjusting its parameters.

B. Seller Reputation

Our seller reputation model aggregates the local values of trust, taking inspiration from the Eigentrust algorithm [11]. The Eigentrust algorithm is one of the most cited and used trust algorithms for peer-to-peer (P2P) systems because it provides an efficient way to build reputation in distributed environments. The original proposal of the Eigentrust algorithm

[1]As in Velloso *et al.* [3], in Malik *et al.* [9], and Putra *et al.* [10].

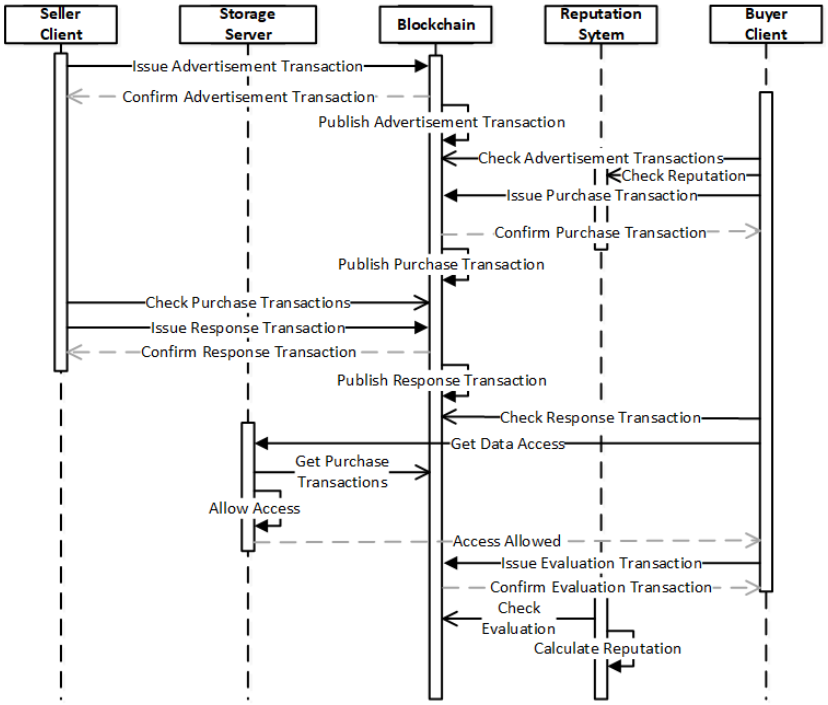


Fig. 2. Sequence diagram of the proposed system representing the data trading process between two organizations.

defines the normalized local trust of a peer *i* in a known peer *j* as:

$$c_{ij} = \frac{max(s_{ij}, 0)}{\sum_j max(s_{ij}, 0)}, \tag{4}$$

where $s_{ij} = sat(i,j) - unsat(i,j)$ is the difference between satisfactory and unsatisfactory interactions amongst the involved peers. Normalization transforms trust values into a probability distribution that prevents a malicious peer from issuing arbitrarily high trust values to other malicious peers. The algorithm then proposes that the natural way for a peer i to discover the reputation of an unknown peer k is to ask his acquaintances about their opinions on k. The opinions of acquaintances have weights proportional to the trust that the peer *i* has in each acquaintance:

$$t_{ik} = \sum_j c_{ij}c_{jk} = c_{i1}c_{1k} + c_{i2}c_{2k} + ... + c_{in}c_{nk}, \tag{5}$$

where t_{ik} represents the trust that peer *i* places in peer *k* based on the opinions of his acquaintances.

Our proposed model replaces, without loss of generality, the original s_{ij} equation with the Gompertz function proposed in (3). The peers i and j that ask for opinions are equivalent to buyers who want to interact with a seller k. Generalizing the equation (5) in the matrix notation for every buyer i, we have:

$$C = \begin{bmatrix} c_{11} & c_{12} & ... & c_{1n} \\ c_{21} & c_{22} & ... & c_{2n} \\ ... & ... & ... & ... \\ c_{n1} & c_{n2} & ... & c_{nn} \end{bmatrix}, \vec{c_{ik}} = \begin{bmatrix} c_{1k} \\ c_{2k} \\ ... \\ c_{nk} \end{bmatrix}, \vec{t_{ik}} = \begin{bmatrix} t_{1k} \\ t_{2k} \\ ... \\ t_{nk} \end{bmatrix} \tag{6}$$

and $\vec{t_{ik}} = C^T.\vec{c_{ik}}$. The trust of buyer *i* in seller k, however, is a limited view based on his own experience and that of his acquaintances. To get a broader view, buyer *i* can ask acquaintances of his acquaintances ($\vec{t_{ik}} = (C^T)^2.\vec{c_{ik}}$). By repeating the process over and over, the buyer gets a complete view of the network after a sufficient number of n iterations:

$$\vec{t_{ik}} = (C^T)^n.\vec{c_{ik}} \tag{7}$$

We highlight that $\vec{c_{ik}}$, C and $\vec{t_{ik}}$ correspond, respectively, to the initial state, the probability matrix and the current state of a stationary Markov chain. Hence, if *n* is large enough and if the matrix C is irreducible and aperiodic, the current state $\vec{t_{ik}}$ of all buyers i will converge to the stationary state $\vec{r_{ik}} = \begin{bmatrix} r_{1k} & r_{2k} & ... & r_{nk} \end{bmatrix}^T$ regardless of the initial trust $\vec{c_{ik}}$ each peer places in seller k. As $r_{1k} = r_{2k} = ... = r_{nk}$, we can then conclude that any element of $\vec{r_{ik}}$ represents the reputation Rep_k of a seller in the system as a whole. Thus, we define a global reputation vector $\vec{r} = \begin{bmatrix} Rep_1 & Rep_2 & ... & Rep_n \end{bmatrix}^T$ that contains the reputation of all sellers in the system and that must be stored in the global state of the blockchain.

C. Data Reputation

Defining the reputation of the data in the proposed system is simpler than the reputation of the seller. We consider

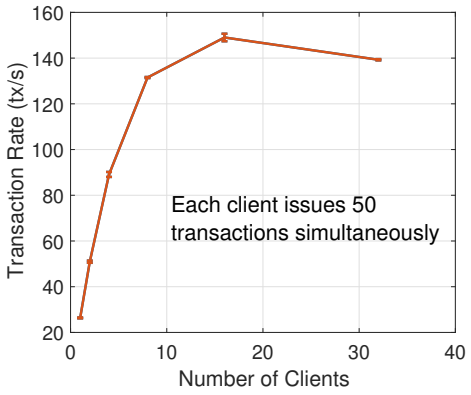


Fig. 3. Transaction rate with an increasing number of clients.

there is no modification of the advertised data because the advertisement transaction contains the hash of the data. A buyer can efficiently detect if a malicious seller changed the advertised data and punish him/her with negative feedback. Hence, it is meaningless to consider dynamic behavior concerning the data. Instead of using ever-changing trust values and a forgetting factor, we can simply define a function based on the number of buyers who have already purchased the data:

$$E_n = \sum_{i=1}^{n} \delta_i \ln(N_b) \frac{Pr}{\sum_{d=m-j}^{m} Pr_d}, \tag{8}$$

where m is the total number of advertisements, Pr is the data price normalized over a window of the most recent system prices, j is the window size, and N_b is the number of users who have already purchased the data. Because $\ln(1) = 0$, the $\ln(N_b)$ factor prevents a malicious seller from increasing his/her reputation by colluding with a single malicious buyer that repeatedly evaluates the seller positively. The factor also improves the reputation of sellers that sell data to many different buyers, thus mitigating cases of small group collusions. We normalize the price to determine its real value compared to the average price in the system. We argue that high-value data hold a high associated risk, and therefore its reputation must increase and decrease proportionally to the risk. We normalize the data price over a window to prevent data reputation from decreasing as more sellers advertise data and, as a consequence, the summation of all advertisement prices in the system increases. The final step is to introduce E_n into the Gompertz function to calculate the reputation of the data:

$$Rep_{dt} = a \exp(-b \exp(-cE_n)). \tag{9}$$

V. Integrating Trust and Reputation into the Blockchain

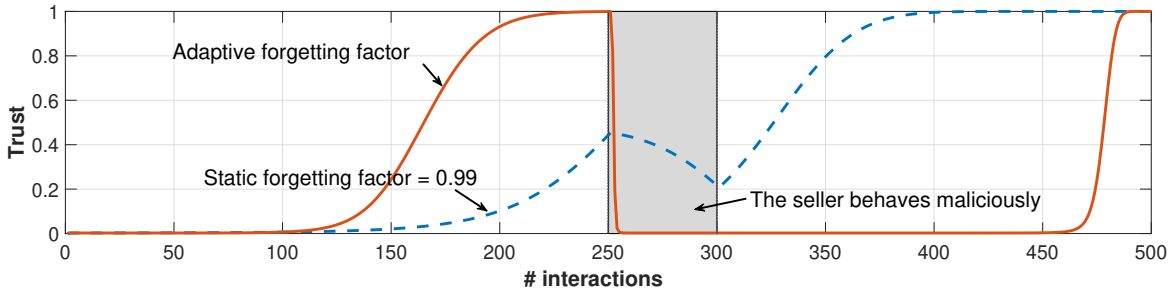
The blockchain has the same structure as the authors used in [2] with the definition of three types of transaction for data trading: i) advertisement transaction; ii) purchase transaction and iii) response transaction. To introduce the trust and reputation system, we propose a feedback in addition to the three transactions. Data owners interested in making their data available and commercialized issue advertisement transactions. The owner submits the advertised data on a storage server capable of supporting and processing large amounts of data [12] and issues a signed advertisement transaction. The system requires that the issuer signs the transactions using asymmetric encryption to guarantee the authenticity and integrity of the transaction, preventing attackers from impersonating the target. The advertisement transaction must contain a brief description of the advertised data type, for example, data from medical sensors, and the data price. Also, the transaction records the advertised data to maintain its integrity.

Buyers search for data in the blockchain by querying advertisement transactions. Those interested in acquiring data advertised in the blockchain issue purchase transactions. The client interested in purchasing the advertised data must issue a purchase transaction referencing the identifier of the corresponding advertisement transaction and informing the buyer's IP address so that an SDN controller can read the transaction and grant access. The purchase transaction must also include the amount to be paid for the data, which will be deducted from the buyer's account once the blockchain logs the transaction. The smart contract associates the purchase transaction with the corresponding advertisement using the transaction identifier and checks if the paid amount is higher than the amount required by the data owner.

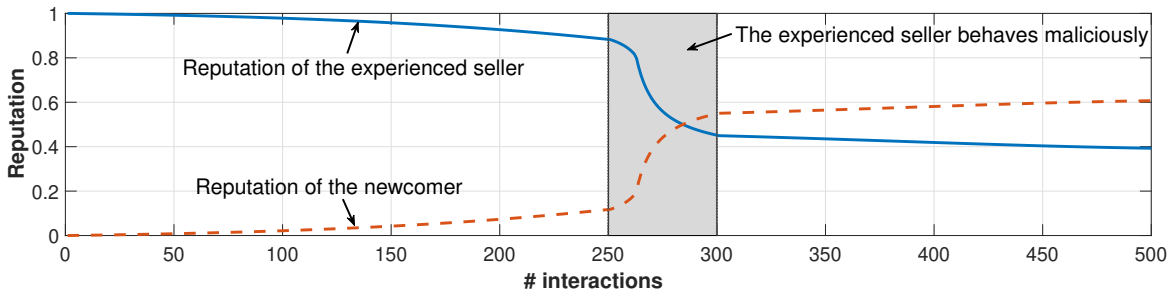
Data owners issue response transactions automatically after a participant buys their data. The response transaction sends the buyer the key that decrypts the data. The key is encrypted using the buyer's public key to ensure that the storage server does not have access to the decrypted data and shares it with third parties. Figure 1 shows the three proposed transactions and their fields to secure data trading.

A smart contract executes the transactions before consensus participants order them in a block. This process consists of the smart contract verifying if the paid amount is greater or equal than the data price requested by the owner and if the organization has enough tokens to make the purchase. Peers validate transactions that meet the previous requirements and invalidate those that do not. If the transaction is valid, the contract deduces the corresponding number of tokens paid from the buyer's organization and sent to the seller's organization, which can use these acquired tokens to buy data from other organizations.

Our proposed system provides trust management in a distributed and autonomous way through smart contracts executed in the blockchain. We propose a feedback transaction TX_{fb} that relies on the smart contract function responsible for calculating a seller's trust and reputation. The feedback transaction contains the corresponding purchase transaction identifier to ensure that the purchase transaction exists and that the buyer needs to sign the evaluation transaction to ensure non-repudiation. After purchasing data d from a seller s, a buyer b can issue a feedback transaction TX_{fb} defined as:



(a) Evolution of a seller's trust considering static and fixed forgetting factor during a on-off attack. The adaptive forgetting factor rewards sellers that have always been legitimate and strongly punishes sellers that have acted malicious, in the range 250 to 300.



(b) Evolution of the reputation of a new seller that receives twice as many positive evaluations as an experienced seller. By acting maliciously in the 250 to 300 range, the experienced seller quickly loses reputation and is overtaken by the newcomer.

Fig. 4. Evolution of the trust and reputation of sellers in the proposed system.

$$TX_{fb} = [TX_{ID_{pur}}|Sig_b|\beta_{b,d_r}|\lambda_{b,s_r}|T_j], \tag{10}$$

where $TX_{ID_{pur}}$ is the corresponding purchase transaction, Sig_b is the buyer's signature, β_{b,d_r} is the buyer's rating for the data d, λ_{b,s_r} is the buyer's rating for the seller s, and T_j is a text field in which the buyer can justify his ratings to the rest of the network. The smart contract defines '0' as a negative interaction and '1' as a positive interaction in the fields β_{b,d_r} and λ_{b,s_r}.

Figure 2 shows the complete data trading process between two domains. If the seller considers the ratings β_{b,d_r} and λ_{b,s_r} unfair, he/she may display a dissatisfaction flag [9]. The smart contract checks if: (i) the seller flags all negative ratings; (ii) the buyer rates all purchase transactions with the seller negatively; (iii) other sellers flagged the buyer. This analysis mitigates bad-mouthing attacks, in which a malicious buyer provides a negative rating to harm an honest seller.

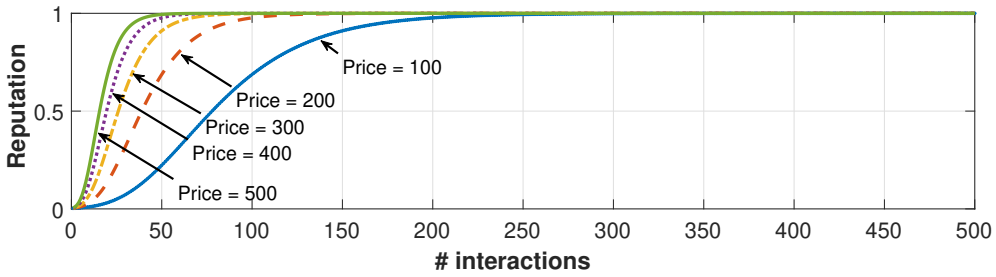
VI. Simulation and Results

We develop a prototype of the proposed system using the open-source platform Hyperledger Fabric v2.0 [13], [14] to implement a permissioned blockchain. The organizational aspect of Hyperledger Fabric fits our proposal multi-domain scenario, in which companies commercialize data. An Intel i7-8700 CPU 3.20 GHz CPU with 32 GB RAM and 12 processing cores deploys the blockchain nodes of the network as Docker containers. We stipulated the number of transactions per block equal to 100, as used in previous work on performance evaluation of the Hyperledger Fabric [15] platform. The Hyperledger Fabric architecture features three types of nodes: clients, peers, and orderers. Clients represent users and issue transactions that need to be executed by endorsers peers, that are responsible for verifying the transaction validity. If the transaction is valid, the client receives the transaction signed by the endorsing peers and sends it with the signatures of the endorsers to ordering nodes, which execute a consensus protocol and order the transactions in a block.

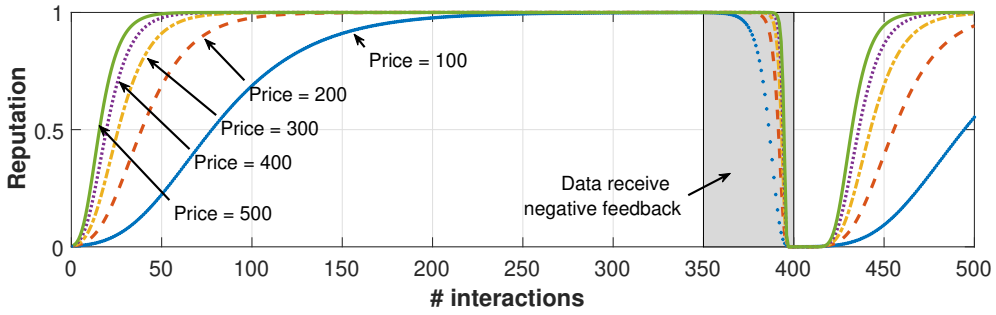
The prototype configuration includes five ordering nodes and the Raft [16] consensus protocol. The smart contract, written in Go, runs on all peers, eliminating a centralized trust entity and implementing the transaction logic[2] described in the previous section, in addition to a system of tokens and organization accounts. Tokens act as currencies that organizations can use to trade data. The real value of these tokens can be agreed off-chain between organizations. The contract restricts the justification field T_j of the feedback transaction TX_{fb} to 280 characters to limit the transaction size.

The first experiment measures the feedback transaction rate issued by users who acquire data from sellers on the blockchain network. The transaction rate corresponds to the ratio between the total number of transactions issued and the time taken for all clients to issue all transactions. Figure 3 shows the experimental results of the transaction rate as a function of the number of clients issuing feedback transactions on the network, with a 95% confidence interval. The rate increases with the number of clients issuing transactions in parallel, as a low number of clients limit the rate of the network to their rate. The system quickly records the feedback of hundreds of clients, reaching an average value of 149

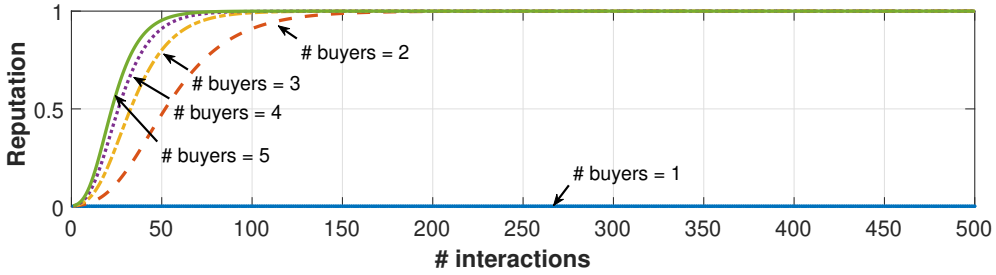
[2]The response transaction TX_{res} will be implemented in future work.



(a) Evolution of data-quality reputation with different advertised prices. The experiment considers only positive evaluations.



(b) Evolution of data-quality reputation with different advertised prices. Upon receiving negative evaluations, the data loses data-quality reputation quickly and needs more positive interactions to regain a high reputation.



(c) Evolution of data-quality reputation with different numbers of buyers. To obtain reputation quickly, the data needs to receive positive evaluations from multiple buyers.

Fig. 5. Evolution of data reputation in the proposed system.

transactions per second.

The second experiment evaluates the impact of the adaptive forgetting factor in comparison to a static forgetting factor when an on-off attack occurs. Figure 4(a) illustrates the three phases of the attack. In the first phase, between interactions 0 and 250, the attacker acts legitimately to gain trust in the system. In this phase, the adaptive forgetting factor rewards his/her good deeds, because, until then, there was no malicious action. In the second phase, between interactions 251 and 300, the attacker takes advantage of the trust built to act maliciously. With a static forgetting factor, the trust of the attacker gradually falls. With an adaptive forgetting factor, the attacker is quickly punished in the first malicious actions until he reaches null trust. In the last phase, between interactions 301 and 500, the attacker returns to behave legitimately to regain his/her trust. With the static forgetting factor, the system quickly forgets the attacker's malicious behavior, which gains trust immediately. With the adaptive forgetfulness factor, however, the attacker remains with zero trust and needs more interactions to recover it again. The experiment demonstrates that, with the adaptive forgetting factor, the system punishes attackers more efficiently and prevents possible damage to several buyers.

The third experiment evaluates the evolution of a newcomer seller's reputation by entering a system where other sellers already own a high reputation. Figure 4(b) illustrates the situation in which a newcomer makes twice as many positive sales as an experienced seller. In the first phase, the new seller starts with a null reputation and gradually increases it as he/she receives more positive interactions than the experienced seller. In this case, the experienced seller, while still having a significantly higher reputation, must increase his positive sales to compete with the new seller or he/she will be slowly overtaken. In the second phase, however, the experienced seller decides to act maliciously by advertising low-quality data to leverage his/her sales number. With the resulting negative interactions, buyers punish the experienced seller, which loses even more reputation and is quickly overtaken by the newcomer. Finally,

in the third phase, the experienced seller returns to the situation of the first phase. Despite being in the system for longer, his reputation is less than that of the new seller. The experiment demonstrates that, just like in real life, the system benefits experienced sellers and requires newcomers to innovate. The experienced seller, however, cannot always rely on experience or act maliciously to preserve his/her reputation, as the new seller will quickly overtake him/her.

The last experiment checks the behavior of data reputation over several interactions. Figure 5(a) illustrates the growth of data reputation according to the advertised price, considering positive interactions only. The experiment considers a window in which the sum of the prices of the most recent sales is equal to 2000 and that the sellers interact with 4 buyers. The growth rate of the data reputation is proportional to the advertised data price, as expected by Equation 8, effectively rewarding high-valued data. In the scenario of Figure 5(b), the data starts to receive negative feedback between interactions 300 and 400 and receives positive feedback after interaction number 400. When receiving 50 negative evaluations, the data reputation in all price ranges reaches the minimum value and, after receiving 50 positive evaluations again, the data does not reach the maximum reputation they had. Therefore, the system effectively punishes malicious behavior regarding data quality. Furthermore, higher-priced data falls at a higher rate than lower-priced data, punishing data reputation proportional to the financial damage caused by the seller. Figure 5(c) illustrates the data reputation growth according to the number of buyers. The experiment considers a window in which the sum of the prices of other more recent sellers is equal to 2000 and that the advertised data costs 400. In this scenario, a seller that interacts with a single buyer keeps the reputation at zero, effectively mitigating collusion between seller and buyer and rewarding sellers who interact with different buyers.

VII. Conclusion

Blockchain technology, combined with smart contracts, provides the necessary transparency for secure data trading, allowing owners to maintain control over their data. The blockchain, however, does not guarantee the delivery of acquired data nor the quality of the advertised data. Therefore, it is necessary to identify the malicious behavior of sellers and make it public to buyers. This paper proposes a secure data marketplace using blockchain, reputation, and trust. Our proposed system allows the commercialization of data automatically, without the need for seller's intervention. Moreover, the paper defines a data reputation that considers the price of the advertised data and builds the seller reputation from the global view of the network on him/her. We implement and evaluate the performance of the proposed system. The results show that the proposed system mitigates traditional attacks from reputation systems efficiently and records transactions quickly in the blockchain, reaching a rate of 150 transactions per second. Our proposed trust and reputation model heavily punishes malicious sellers and poor data quality. As future work, we intend to implement the proposed reputation system in smart contracts to guarantee the reputation processing in a distributed and automatic way.

VIII. Acknowledgment

This work was financed by XXXX, XXXXX, and XXXXXX

References

[1] E. Lodderstedt, M. McGloin, and P. Hunt, "OAuth 2.0 threat model and security considerations," 2013, iETF. RFC 6819. Available at http://www.rfc-editor.org/rfc/rfc6819.txt. Last access: 15 July 2020.

[2] Anonymous, 2020.

[3] P. B. Velloso, R. P. Laufer, D. de Oliveira Cunha, O. C. M. B. Duarte, and G. Pujolle, "Trust management in mobile ad hoc networks using a scalable maturity-based model," *IEEE Transactions on Network and Service Management*, vol. 7, no. 3, pp. 172–185, Sep. 2010. [Online]. Available: https://ieeexplore.ieee.org/document/5560572/

[4] Y. Sun, Z. Han, and K. R. Liu, "Defense of trust management vulnerabilities in distributed networks," *IEEE Communications Magazine*, vol. 46, no. 2, pp. 112–119, Feb. 2008. [Online]. Available: http://ieeexplore.ieee.org/document/4473092/

[5] G. A. F. Rebello, I. D. Alvarenga, I. J. Sanz, and O. C. M. B. Duarte, "BSec-NFVO: A blockchain-based security for network function virtualization orchestration," in *IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[6] R. A. Michelin, A. Dorri, R. C. Lunardi, M. Steger, S. S. Kanhere, R. Jurdak, and A. F. Zorzo, "SpeedyChain: A framework for decoupling data from blockchain for smart cities," in *Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 2018, pp. 145–154.

[7] M. T. de Oliveira, L. H. Reis, D. S. Medeiros, R. C. Carrano, S. D. Olabarriaga, and D. M. Mattos, "Blockchain reputation-based consensus: A scalable and resilient mechanism for distributed mistrusting applications," *Computer Networks*, vol. 179, p. 107367, Oct. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1389128620300360

[8] R. Dennis and G. Owen, "Rep on the block: A next generation reputation system based on the blockchain," in *ICITST'2015*. London: IEEE, Dec. 2015, pp. 131–138. [Online]. Available: http://ieeexplore.ieee.org/document/7412073/

[9] S. Malik, V. Dedeoglu, S. S. Kanhere, and R. Jurdak, "TrustChain: Trust Management in Blockchain and IoT Supported Supply Chains," in *IEEE Blockchain'2019*. Atlanta, GA, USA: IEEE, Jul. 2019, pp. 184–193. [Online]. Available: https://ieeexplore.ieee.org/document/8946187/

[10] G. D. Putra, V. Dedeoglu, S. S. Kanhere, and R. Jurdak, "Trust Management in Decentralized IoT Access Control System," in *IEEE ICBC'2020*. IEEE, 2020, p. 9.

[11] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The Eigentrust algorithm for reputation management in P2P networks," in *Proceedings of the WWW'03*. Budapest, Hungary: ACM, May 2003, pp. 640–651. [Online]. Available: https://doi.org/10.1145/775152.775242

[12] H. T. T. Truong, M. Almeida, G. Karame, and C. Soriente, "Towards secure and decentralized sharing of IoT data," in *IEEE Blockchain'2019*, 2019, pp. 176–183.

[13] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger Fabric: A distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference*. ACM, 2018, p. 30.

[14] T. S. L. Nguyen, G. Jourjon, M. Potop-Butucaru, and K. L. Thai, "Impact of network delays on Hyperledger Fabric," in *IEEE Conference on Computer Communications Workshops (INFOCOM)*, 2019, pp. 222–227.

[15] C. Gorenflo, S. Lee, L. Golab, and S. Keshav, "FastFabric: Scaling hyperledger fabric to 20,000 transactions per second," in *IEEE ICBC'2019*, 2019, pp. 455–463.

[16] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *USENIX ATC'2014)*. Philadelphia, PA: USENIX, Jun. 2014, pp. 305–319. [Online]. Available: https://www.usenix.org/conference/atc14/technical-sessions/presentation/ongaro