DARS: Empowering Trust in Blockchain-Based Real-World Applications with A Decentralized Anonymous Reputation System

Mouhamed Amine Bouchiha, Yacine Ghamri-Doudane, Mourad Rabah, Ronan Champagnat

DARS: Empowering Trust in Blockchain-Based Real-World Applications with A Decentralized Anonymous Reputation System

Mouhamed Amine Bouchiha, Yacine Ghamri-Doudane, Mourad Rabah and Ronan Champagnat

Abstract Blockchain-based Reputation Systems (BRS) are a recent and essential development in decentralized trust and reputation management. The decentralization, transparency, and efficiency brought by Blockchain (BC) are clearly what we always hoped for to build effective trustless reputation systems. Despite these promising attributes, existing BRS face a critical challenge in countering common reputation attacks, including whitewashing, self-promotion, and bad-collision attacks. Currently, BRS rely on reputation scores or tokens linked to the same address or public key, which severely limits their widespread adoption as it raises concerns about possible retaliation, hence the reluctance of users to engage and provide feedback. In this work, we propose and develop a Decentralized Anonymous Reputation System (DARS) for trust-related applications. In DARS, users can use different pseudonyms when interacting with each other to hide their digital identities. In our system design, all pseudonyms of a specific user, yet, are cryptographically linked to the same access token, allowing honest users to maintain their reputation and preventing malicious ones from starting over. This is achieved through the use of zkSNARK proofs for set membership via Merkle trees over commitments. We extended our framework with an efficient reputation model that respects all the security and privacy properties of our formal model. Finally, we developed a prototype of the proposed framework using emerging technologies and cryptographic tools. The evaluation results demonstrate the feasibility and effectiveness of DARS.

Paper accepted at the 38th International Conference on Advanced Information Networking and Applications (AINA-2024) Kitakyushu International Convention Center, Kitakyushu, Japan, April 17 to April 19, 2024 and will be published by Springer in the conference proceedings of Lecture Notes in Data Engineering and Communication Technologies.

Mouhamed Amine Bouchiha, Yacine Ghamri-Doudane, Mourad Rabah, and Ronan Champagnat
L3i Laboratory, La Rochelle University. e-mail: {mouhamed.bouchiha,yacine.ghamri,mourad.rabah,ronan.champagnat}@univ-lr.fr

1 Introduction

Reputation, referencing the overall opinion towards a user or an entity, gained widespread adoption since its inception [1]. Reputation Systems (RS) aim to hold users accountable for their behaviors [2]. While traditional models present the central server as semi-honest, maintaining user privacy becomes uncertain if it becomes malicious due to external or internal compromises [3]. Despite cryptographic approaches addressing some security issues, the single point of failure and the lack of transparency remain significant challenges for centralized reputation systems [4].

Reputation systems often involve gathering and analyzing user data to determine and display reputation scores. This aspect can give rise to privacy concerns, particularly if users feel uncomfortable about sharing personal information or activity data. Addressing worries of potential retaliation, which often deter users from engaging and offering feedback, can involve employing feedback-independent reputation models [5]. Nevertheless, the persisting issue of linking reputation scores and tokens to a single master key remains a cause for concern regarding potential tracking. A recent approach to address this issue involves the development of decentralized privacy-preserving reputation models, allowing users to interact and share feedback confidentially and seamlessly. This advancement stands as a significant leap in trust and reputation management, offering both robust reputation management and user privacy preservation. The use of cryptographic techniques with decentralized systems such as Blockchain [10, 11] can help reputation systems guarantee user privacy without compromising transparency and efficiency. However, existing solutions fall short of being entirely decentralized since they depend either on a centralized entity or a group of trusted peers to handle identities, credentials, and security parameters. Additionally, despite the use of BC technologies in numerous research efforts [13–15] to develop decentralized and privacy-centric reputation systems, these proposed solutions fail to sufficiently tackle the efficiency challenges of on-chain reputation management. Furthermore, the issues associated with the implementation of real-world blockchain-based reputation frameworks, particularly the Oracle problem [18], have not undergone thorough examination. Therefore, to overcome all these issues, we propose in this paper, "DARS", a fully Decentralized Anonymous Reputation System for real-world applications. The main contributions of this research include:

- A decentralized reputation system that is constructed on top of two distinct ledgers to separate identity management from business activities.
- A system that relies on Decentralized Oracle Networks not only to automate smart contracts execution but also to import credentials from existing systems to prevent Sybil attacks.
- The use of zkSNARK proofs for Set Membership over commitments, allowing DARS users to gain the ability to generate and use numerous pseudonyms to safeguard their digital identity and ensure anonymity.
- A design of a reputation model that achieves all the security and privacy properties of our formal model.

- A proof-of-concept for the proposed framework, leveraging emerging technologies and cryptographic tools is developed. This allows for a more meaningful assessment of DARS's capabilities.

The remainder of this paper is structured as follows: Section 2 describes related work. The security model is presented in section 3. Section 4 deals with cryptographic building blocks. Section 5 is devoted to the construction of the proposed Decentralized Anonymous Reputation System (DARS). Section 6 is dedicated to the performance evaluation of the proposal and Section 7 to the security analysis.

2 Related work

Blockchain(BC)-based reputation systems are now essential for trust-based applications such as retail marketing, mobile crowdsensing, and decentralized markets. Considerable research has focused on developing anonymous and privacy-preserving reputation systems for these areas. These systems make it challenging to link a user's actions/feedback with their true identity. However, maintaining an accurate reputation without any ties to identity raises difficulties, as the reputation score should reflect precisely the user's activity within the system. In an attempt to bring an answer to the above concern Liu et al. [10] propose an anonymous reputation system for retail marketing in the industrial IoT environment. The system utilizes smart contracts (SCs) on a PoS-based BC, ensuring transparency and public verifiability even against malicious attacks. It prioritizes anonymity through randomizable signatures and zero-knowledge proofs (ZKPs). However, the system's reliance on centralized IDentity Management (IDM) for managing identities, credentials, and security parameters creates a potential security risk and vulnerability to a single point of failure. [11] introduces a privacy-focused reputation system using BC in mobile crowdsensing with limited resources. Reputation scores are updated globally using SCs through feedback averages. The system employs additive secret sharing and delegation sets for privacy in a dynamic setting. However, like prior research, it operates under a semi-honest model, leading to potential security concerns. BPRF [12] presents a BC-based privacy-preserving reputation framework for participatory sensing systems. The system uses SCs to manage the reputation scores of participants based on their sensing data and corresponding feedback. The solution employs group signatures and a partially blind signature algorithm to protect user information. To achieve greater transparency, Schaub et al. [13] proposed a fully decentralized reputation system on top of a public BC with a blind signature to guarantee consumer anonymity. Soska et al. [14] proposed an anonymous reputation system based on a ring signature, which resulted in linear overhead when generating the anonymous review proof. Although, [12–15] have made significant efforts to investigate the use of BC technology in constructing robust privacy-preserving reputation systems, the proposed solutions have not fully examined the efficiency and scalability issues associated with BC-based reputation management. Moreover, the implementation challenges specific to BC-based solutions like the Oracle prob-

lem [18], have not been explored in depth in these studies. To overcome all the above issues, we propose in this paper, "DARS", a decentralized anonymous reputation system for BC-based real-world applications.

3 Security Model

In this section, we introduce the adversarial model and explore the security features of the proposed DARS.

3.1 Adversarial model

For our adversarial model, we borrow the assumptions of [7]. Therefore, an adversary is able to statically and actively corrupt up to t of the n nodes in the committee, for $t < n/3$. In addition, the adversary can corrupt any number of external entities, such as users and applications.

3.2 Security properties

Under the above assumptions, we outline the security properties and objectives of DARS as follows:

- **Sybil resistance:** A user cannot have any credentials other than his/her own.
- **Unforgeability:** An adversary cannot forge the credentials of honest users or impersonate them.
- **User privacy**: It is infeasible for an adversary to ascertain a user's attributes through the examination of issued identification information, the analysis of transaction data during interactions with other users, or the observation of the ongoing evaluation of interactions.
- **Reputation binding:** The user's reputation is unique and stored publicly in the BC. Although users can generate as many pseudonyms as they wish, all are cryptographically linked to the same access token.
- **Forward Reputation binding:** No user should be able to mint/use a reputation token with a reputation score higher than that linked to his/her last token.

4 DARS Building Blocks

In this section, the main cryptographic building blocks upon which the DARS system is built are presented.

4.1 Commitment scheme

A commitment scheme is a cryptographic protocol that allows a party, referred to as the committer, to commit to a chosen value without revealing it, while still being able to prove its validity later on [19]. It is designed to fulfill two crucial security properties: (*i*) Hiding Property: Given $\mathsf{COMM}(\mathsf{x})$, it should be computationally infeasible to determine the original value x. (*ii*) Binding Property: It should be computationally infeasible to find two distinct values x_1 and x_2 such that $\mathsf{COMM}(\mathsf{x}_1) = \mathsf{COMM}(\mathsf{x}_2)$.

4.2 zkSNARKs

Zero-Knowledge Succinct Non-Interactive Argument of Knowledge (zkSNARKs) are an advanced variant of Zero-Knowledge Proofs (ZKPs). More precisely, a zkSNARK scheme is a Non-Interactive Zero-Knowledge (NIZK) scheme [8], wherein the proof itself is a self-contained data block that can be verified without requiring any interaction from the prover [20,21]. A zkSNARK construction consists of three algorithms (Gen, Prov, Verif) defined as follows:

- The key generator G takes a secret parameter λ and a program C, and generates two publicly available keys: a proving key pk, and a verification key vk. These keys are public parameters that need to be generated only once for a given program C.
- The prover P takes as input the proving key pk, a public input t, and a private witness w. The algorithm generates a proof $\pi = \mathsf{Prov}(pk,t,w)$ that the prover knows a witness w and that the witness satisfies the program C.
- The verifier V computes $\mathsf{Verif}(vk,t,\pi)$ which returns true if the proof is correct, and false otherwise. Thus, this function returns true if the prover knows a witness w satisfying C.

4.3 zkSNARKS for Set Membership via Merkle Trees

The set membership problem via Merkle trees involves proving that an element belongs to a set using the Merkle tree data structure. More formally, Given a set S containing n elements and a Merkle tree constructed from the hash values of these elements, the set membership problem is to prove that a specific element x belongs to the set S without revealing any other elements in S [9]. Merkle trees alone do not provide ZK property. To achieve this property, we need to combine Merkle trees with additional techniques such as zkSNARK or other cryptographic primitives [16].

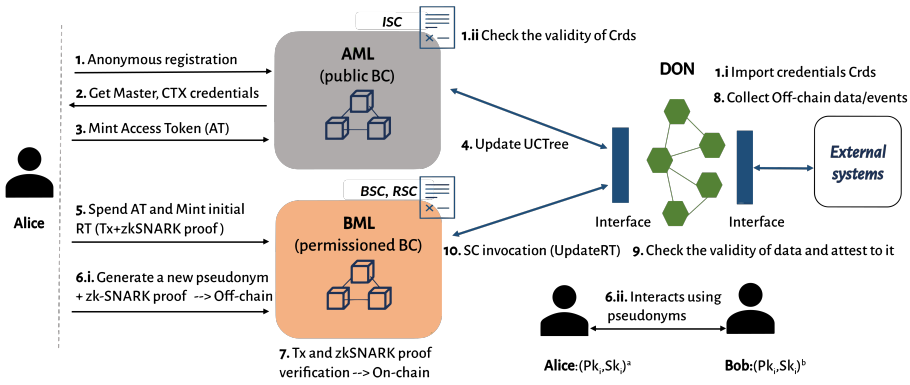


Fig. 1: Overview of DARS Framework, comprising an Access Management Ledger (AML), a Business Management Ledger (BML), and a Decentralized Oracle Network (DON).

5 DARS Framework

In this section, we describe our proposed Decentralized Anonymous Reputation System (DARS). An overview of DARS is shown in Figure 1. We use the previously mentioned building blocks on top of two separate ledgers, *Access Management Ledger (AML)* and *Business Management Ledger (BML)*, to decouple identity management from business operations. AML is a public permissionless BC responsible for managing identities and access, while BML is a permissioned BC that implements the overall business logic of a real-world application. Our construction aims to ensure robust and efficient reputation management over different pseudonyms. We build our framework over four main phases:

Phase1-Registration. The registration process takes place between a user u and the AML committee. The AML relies on a Decentralized Oracle Network (DON) to ensure uniqueness while issuing master credentials M_{cred} for any valid user u. As in [7], we make use of DON to import identities from existing systems. For example, Alice can use her credentials on her Social Security Administration (SSA) account to generate a credential certifying her Social Security Number (SSN). Our DON uses the DECO protocol [17] to provide privacy for user data. DECO is a three-party protocol involving a prover denoted as P, a verifier denoted as V," and a TLS server denoted as S. The protocol enables P to persuade V that a data item, which may be private to P, obtained from S, meets a specified Predicate. DECO relies on multiparty computation (MPC) to protect the confidentiality and authenticity of the data, and on zero-knowledge proofs (ZKPs) to prove that a predicate is satisfied.
Once the identity of the user u is verified by the committee nodes following the DECO protocol, the corresponding user will be able to post his set of claims on AML and get access on BML using context-based credentials. To obtain a new credential for the ctx context, *e.g.* "a trading or crowdsourcing activity", u must submit to the committee $(pk_{ctx}, M_{cred}, C^u_{ctx})$: a new identifier to be used in the ctx

context, master credentials and a set of pre-credentials with the claims C^u_{ctx} required by ctx. The committee upholds a set of $Granted_{ctx}$ identifiers denoting those that have already obtained a credential within this specific context. If M_{pk^u} of M_{cred} is not part of this set, a credential is issued. Finally, (M_{pk^u}, pk^u_{ctx}) is added to $Granted_{ctx}$. For more details on issuing context-based credentials, see [7].
In our construction, master credentials are purposely excluded from any interactions with the BML to prevent linking them to the user's real identity. On the other hand, contextual credentials are used exclusively on the AML and remain cryptographically hidden on the BML to separate identity management from business operations. To guarantee these properties, the committee nodes maintain a CRH-based Merkle tree with root rt called $UCTree$, which contains all the user access commitments. When a user u is successfully registered with the committee nodes he/she must provide a commitment to his/her credentials. The user will use this commitment to access the BML without revealing any information that could be linked to his/her identity. To do that the user proceeds as follows: u generates an address key pair (apk, ask), the address public key and private key, respectively; u samples a random a and computes $cm_u = \mathsf{COMM}_a(pk^u_{ctx})$, then computes $cm_A := \mathsf{COMM}_b(cm_u||apk)$ for a secret b, and defines $AT := (pk^u_{ctx}, a, b, cm_A)$. A corresponding mint access token transaction, $tx_{AM} := (apk, cm_u, cm_A)$, is added to the AML (accepted only if pk^u_{ctx} is known to the committee). The $UCTree$ is then updated with a new leaf (cm_A). We use DON to synchronize any changes made to the $UCTree$ on the AML with the BML. This construction allows the user to prove to BML validators that he/she has valid credentials on AML efficiently and anonymously, *i.e.* the time and space complexity is logarithmic to the size of $UCTree$, and pk^u_{ctx} remains cryptographically hidden in cm_u.

Phase2-User anonymity with a reputation token. The second phase in our construction is the mint of a reputation token that is cryptographically linked to the user's contextual credentials. This phase aims to hide the user's digital identity pk^u_{ctx} while ensuring robust reputation management through the utilization of zkSNARK proofs and a commitment scheme.
To interact with other users and post transactions on the BML, the user must spend their access tokens and mint their initial reputation token RT_{init}. This is equivalent to adding a new leaf to the $RCTree$ (similar to $UCTree$) containing a commitment to its initial reputation score. To achieve the property of forward reputation binding the user must sample a random serial number S_n for each new reputation token. S_n is then released when using the token. This is realized as follows, u first generates a new key pair (pk, sk), then samples a random s and computes $S_n := \mathsf{PRF}_{sk}(s)$ using a Pseudo-Random Function (PRF), and commits to the tuple (pk, R_{init}, s) in two steps: $cm_P := \mathsf{COMM}_r(pk||s)$ for a random r, and then $cm_R := \mathsf{COMM}_{r'}(R_{init}||cm_P)$ for another random r'. The outcomes comprise: (i) a reputation token $RT_{init} := (pk, R_{init}, s, r, r', cm_R)$ and (ii) a RT mint transaction $tx_{RM} := (R_{init}, cm_P, r', cm_R)$. However, this alone does not fulfill the criteria for the transaction to gain acceptance on the BML. The user must provide a zkSNARK proof π_A of the NP statement ***"I know a secret*** b ***such that*** $\mathsf{COMM}_b(cm_u||apk)$ ***ap-***

pears as a leaf in a CRH-based Merkle tree $UCTree$ ***whose root is*** rt***"***. This prerequisite permits access to the BML exclusively for authorized users. With this in mind, we edit the tx_{RM} transaction to $tx_{RM} := (R_{init}, cm_P, r', cm_R, \pi_A)$ which is submitted to the BML. The tx_{RM} is accepted if and only if the π_A and cm_R are valid. Because of commitment nesting, anyone can verify that cm_R in tx_{RM} is a commitment of a token of value R_{init} (by checking that $\mathsf{COMM}_{r'}(R_{init}||cm_P)$ equals cm_R), but is unable to identify the owner through the knowledge of the address key pk or the serial number S_n (derived from s), as these are hidden in cm_P. Finally, user anonymity is achieved because the proof π_A is zero-knowledge: while cm_u and apk are revealed, no information about b is revealed, and finding which of the many commitments in $UCTree$ corresponds to tx_{RM} is equivalent to inverting $f(b) := \mathsf{COMM}_b(X)$, which is assumed to be infeasible [16].

Phase3-Reputation token use/spending. So far, user u has minted his initial reputation token RT_{init}. He can therefore interact with any user v on the BML by submitting transactions. Within DARS, users' reputation scores are tied to their most recent reputation commitment cm_R. As a result, for a user u to engage with other users, they must reveal the nested commitment to display their reputation score. Since only the user possessing the secret r' can unveil it, there's no susceptibility to forgery. Additionally, tx_{RM} is submitted using a pseudonym different from pk^u_{ctx}, and since u can generate numerous pseudonyms (ideally, a new pseudonym apk^{new} for each new interaction), the likelihood of disclosing its actual identity is effectively eliminated.

Let's delve deeper into the details. Users within the BML can utilize their reputation token through the submission of a reputation spending transaction, denoted as tx_{spend}. This transaction enables them to create a new token of identical value to the current one. Consider a scenario where a user u possesses a pair of address keys (apk^{old}, ask^{old}), wishes to consume its current token $RT^{old} := (apk^{old}, R^{old}, s^{old}, r^{old}, r'^{old}, cm_R^{old})$ and produce a new one RT^{new}, targeted at the public address key apk^{new}. The user u proceeds as follows, (i) u samples serial number randomness s^{new}; (ii) u computes $cm_P^{new} := \mathsf{COMM}_r^{new}(apk^{new}||s^{new})$ for a random r^{new}; and then computes (iii) $cm_R^{new} := \mathsf{COMM}_{r'^{new}}(R^{new}||cm_P^{new})$ for a random r'^{new}. This yields the token $RT^{new} := (apk^{new}, R^{new}, s^{new}, r^{new}, r'^{new}, cm_R^{new})$. Next, u generates a zkSNARK proof π_{RS} for the following NP statement:

"Given the $RCTree$ ***root*** rt_R***, serial number*** S_n^{old}***, and token commitment*** cm_R^{new}***, I know a token*** RT^{old}***,*** RT^{new}***, and address secret key*** sk^{old} ***such that:***

(i) The tokens are well-formed: $cm_P^{old} := \mathsf{COMM}_{r^{old}}(apk^{old}||s^{old})$ and $cm_R^{old} := \mathsf{COMM}_{r'^{old}}(R^{old}||cm_P^{old})$ for cm_R^{old} and similarly for cm_R^{new}.
(ii) The secret key matches the public key: $apk^{old} = \mathsf{PRF}_{ask^{old}}(0)$.
(iii) The serial number is calculated correctly: $S_n^{old} = \mathsf{PRF}_{ask^{old}}(s^{old})$.
(iv) The commitment cm_R^{old} appears as a leaf in $RCTree$ whose root is rt_R.
(v) The reputation values are equal $R^{new} = R^{old}$."

A resulting spend transaction $tx_{RS} := (rt_R, S_n^{old}, cm_R^{new}, \pi_{RS})$ is sent to the BML. tx_{RS} gets rejected if S_n^{old} appears in a prior transaction. Thus, the user is forced to use

his/her most recent reputation token for each new interaction.

Phase4-Reputation Update. This process is done automatically using the DON and smart contracts. We use the DON to collect the off-chain data needed to evaluate the interaction, and then trigger the reputation module implemented using smart contracts to update the reputation scores of the users involved in the interaction using their pseudonyms. The update process takes place once the interaction is over. For a more comprehensive explanation, let's consider a scenario within a marketplace. Let's suppose that user u wants to introduce a new product into the system. In this case, u is faced with two choices: use the existing reputation token or mint a new one, as detailed earlier. Then, u can simply add the new product to the system by posting the corresponding transaction $tx_{newProd} := (prodID, price, DescriptionpathonIPFS, ...)$. Once the product is listed for sale, when another user v, intends to purchase this item from u, v has the option to either utilize their existing reputation token or spend it to generate a new one, retaining the same reputation score and ensuring ongoing anonymity. Additionally, v is required to provide a zkSNARK proof demonstrating the absence of a shared secret key with u. This condition is necessary for our construction, as it prevents self-promotion attacks. To do this, v computes $H_i^v := H(prodID||sk_{ctx}^v)$. Then produces a zkSNARK proof π_i^v for the following NP statement, ***"Given the product identifier prodID, I know*** sk_{ctx}^v, cm_R ***, and*** RT_i ***such that,*** H_i^v ***is computed correctly and shares the same*** sk_{ctx}^v ***with*** apk_i***"***. The proof is then sent to the BML as part of the new order transaction $tx_{newOrd} := (ordID, info, H_i^v, \pi_i)$. Like v, if u chooses to approve v's order, u is required to compute the interaction hash $H_i^u := H(prodID||sk_{ctx}^u)$ using its own sk_{ctx}^u and provide the corresponding zkSNARK proof. The proof is then sent to BML as part of the order acceptance transaction $tx_{accOrd} := (ordID, info, H_i^u, \pi_i^u)$. The transaction is rejected if H_i^u and H_i^v are identical or if the proof π_i^u is invalid.
Once the off-chain interaction is over, users u and v must transmit the data needed to evaluate the interaction. In our system design, we use DON to collect data from external systems, verify it, and calculate the required values of all the metrics used in the reputation model. Then, the Reputation Smart Contract (RSC) is triggered to perform the evaluation and update the global reputation scores. Both u and v have shown their last reputation commitments using their pseudonyms apk_i^u and apk_i^v, respectively. Consequently, the RSC will update the reputation scores of u and v automatically and transparently using the revealed information. It's crucial to note that the evaluation of the interaction itself should prioritize privacy, ensuring that our reputation model doesn't utilize or disclose any details regarding the users' identities engaged in the interaction. To achieve this, we propose employing the following formula for interaction evaluation:

$$\begin{cases} \mathbf{T}_i = \mathscr{P}\ [\omega_p + \omega_t F_t + \omega_a F_a] \\ \omega_p, \omega_t, \omega_a \in [0,1]\ ;\ \omega_p + \omega_t + \omega_a = 1 \end{cases} \tag{1}$$

where $\mathscr{P}$ is a Boolean which refers to the presence of the proof "1" or not "0", *i.e.* whether the interaction outside the chain has actually taken place or not. This could be proof of delivering a "product" or completing a "task". ω_p is the weight of the

proof itself. ω_t and ω_a are the weights of the time t and the amount a of the interaction, respectively. F_t and F_a are the functions that normalize t and a, respectively ($F_a, F_t \in [0,1]$). The value and timing metrics are implemented to thwart coordinated attacks. These attacks occur when users collude to boost each other's reputation by engaging in multiple low-cost interactions within a brief timeframe. The formula can be extended with additional contextual factors (feedback, data quality...), provided they do not reveal any information about the user's digital or physical identity. The global reputation update is performed using the following formula [5]:

$$R_{new} = \begin{cases} (1-\mathscr{W}_f)R_{old} + \mathscr{W}_f \mathbf{T}_i \ ; \mathbf{T}_i \geq T_{min} \\ \mathscr{W}_f R_{old} + (1-\mathscr{W}_f)\mathbf{T}_i \ ; \mathbf{T}_i < T_{min} \end{cases} \quad (2)$$

where R_{old} is the old reputation, $\mathbf{T}_i$ is the value of the interaction, T_{min} is the trust threshold, and $\mathscr{W}_f \in [0,1]$ is a weighting function that gives more or less relevance to $\mathbf{T}_i$, depending on the role played by the user in the interaction, *e.g.* "seller" or "buyer", and the value of the interaction itself (*i.e.* positive or negative interaction).

6 Evaluation and Results

In this section, we first introduce the evaluation environment and experimental setup for the proposed DARS. We then discuss the on/off-chain evaluation results.

6.1 Evaluation Environment

We carried out the benchmarks on our local BC platform. The platform is a cluster of two HPE ProLiant XL225n Gen10 Plus servers dedicated to the experimentation and evaluation of BC solutions. Each server features two AMD EPYC 7713 64-Core 2GHz processors and 2x256GB RAM.

6.2 Experimental Setup

To evaluate the proposed solution, we developed a proof of concept for the Decentralized Anonymous Reputation System (DARS) by leveraging cutting-edge technologies and cryptographic tools. The circuits employed in DARS are implemented using the circom programming language and the circomlib library[1]. We utilized the snarkjs library[2] to compile the circuits and perform the powers of tau ceremony for the trusted setup. Additionally, we developed the smart contracts of DARS using

[1] https://github.com/iden3/circomlib

[2] https://github.com/iden3/snarkjs

the Solidity programming language[3] and established a local network consisting of twelve validators using Hyperledger Besu[4] as BC client with Proof of Authority (PoA) as consensus protocol. We utilized Web3js library[5] for developing the client side and deploying the system's smart contracts. Lastly, for conducting benchmarking tests, we utilized Hyperledger Caliper [6].

6.3 Performance Evaluation

Three metrics are considered for DARS performance evaluation:

- **Time overhead:** refers to the processing time for the proving and verification operations. This time is measured off-chain for the proving operation; or from when a specific transaction that contains a zkSNARK proof is received at the smart contract (on-chain) for verification until the appropriate response is sent back to the prover.
- **Throughput:** refers to the number of successful transactions per second (TPS).
- **Latency:** is the time difference in seconds between the submission and completion of a transaction.

Table 1: Time overhead measurements for the zkSNARK proofs generation and verification using the Groth16 proving system.

Tx type	Proving(ms)	Verification(ms)	Overall Time(ms)	Call Data size
spendAT (π_A)	2400	730	3130	705B
spendRT (π_{RS})	2900	950	3850	705B
spendRT (π_{RS})	480	640	1120	705B

Table 2: Time overhead measurements for zkSNARKS proofs generation and verification using the PlonK proving system.

Tx type	Proving(ms)	Verification(ms)	Overall Time(ms)	Call Data size
spendAT (π_A)	67000	760	67760	1750B
spendRT (π_{RS})	79500	935	80435	1750B
newOrd (π_i)	3400	670	4070	1750B

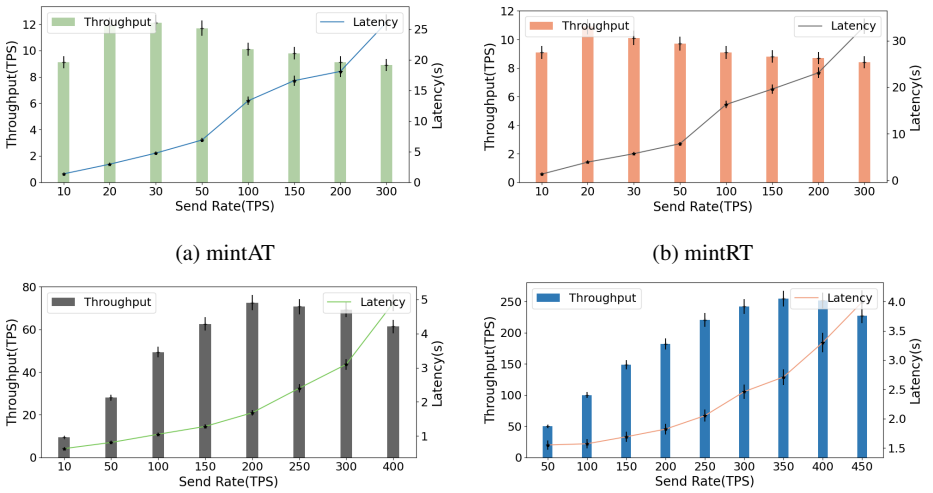
A. Time Overhead. We employed two distinct proving systems, namely zkSNARK Groth16 and Plonk to evaluate the time overhead of our circuits. Groth16 is a circuit-

[3] https://docs.soliditylang.org

[4] https://besu.hyperledger.org

[5] https://web3js.readthedocs.io

[6] https://github.com/hyperledger/caliper-benchmarks



(a) mintAT (b) mintRT

(c) spendRT (d) updateRT

Fig. 2: Latency and Throughput of DARS under different send rates

specific preprocessing general-purpose zkSNARK construction that has become a standard choice in various BC projects [20]. This popularity is owed to its proofs' constant size and efficient verifier time. However, Groth16 necessitates a circuit-specific trusted setup during its preprocessing phase, which could be considered a drawback. On the other hand, PlonK represents a universal preprocessing general-purpose zkSNARK construction [21]. This proving scheme features an updatable preprocessing phase and boasts a short and constant verification time. Nevertheless, PlonK proofs tend to be larger and take more time (60-80s) to generate compared to Groth16. Table 1 presents the timing and memory-related measurements for the Groth16 ZK-proof components, namely π_A, π_{RS}, and π_i. The π_A allows proving the existence of valid AML credentials efficiently and anonymously, π_{RS} proves the validity of RT, and π_i attests the validity of the interaction, preventing self-promotion attacks (see Sec. 5). Additionally, Table 2 displays the corresponding measurements utilizing the PlonK construction. Experimental results show that proof generation and verification take only a few milliseconds (480-2900ms) when using the Groth16 scheme, whereas proof generation with the PlonK system takes a relatively longer time (3.4-80s). Compared to verification with Groth16, no significant difference is observed for the proof verification using PlonK.

B. Throughput and Latency. We conducted a series of experiments using Caliper to qualitatively evaluate DARS' performance. The experiments involved changing the Tx sending rate (ranging from 10 to 500 TPS) using a consistent network configuration for the four main operations performed within our system. The results are illustrated in Figure 2. As the Tx sending rate increases for each operation, the

throughput also increases accordingly. Regarding the updateRT Tx, it reaches a peak of $255TPS$ with a sending rate of $350TPS$, and then experiences a decline, indicating an overloaded system. On the other hand, system latency for the updateRT Tx remains relatively small and stable (less than $3s$), as long as the system is not overloaded. The remaining operations exhibit similar behavior, but their performances are comparatively lower. Notably, the mintAT and mintRT operations stand out as the most computationally intensive due to the substantial amount of computation needed to insert the commitments cm_A and cm_R into the $UCTree$ and $RCTree$ structures, respectively. This heightened computational intensity directly contributes to the heavier workload experienced by these operations. It is essential to highlight that all operations are currently sent directly to the main chain without employing any scaling solution. Indeed, high scalability was not our primary objective in this paper. Though the achieved scalability remains competitive.

7 Security Analysis

In this section, we examine the key security risks and the measures implemented by DARS to counter these threats.

- **Sybil Attacks:** involve creating multiple pseudonymous to manipulate the reputation system. In DARS, each legitimate user is granted only one valid credential for each specific context. Specifically, the AML committee maintains a set of $Granted_{ctx}$ identifiers, representing those that have already received a credential within that context. If M_{pk^u} in M_{cred} is not part of this set, a credential is granted; otherwise, no additional credential is issued for that user. This design ensures that a user cannot generate and utilize more than one valid access token per context. Consequently, DARS effectively guards against Sybil attacks.
- **Unforgeability:** Identity Theft is mitigated in the AML subsystem as users' keys remain in their wallets. These keys are utilized only for signing challenges during the protocol as part of credential verification. Consequently, the assurance of unforgeability within this subsystem is a direct consequence of the overall unforgeability of signatures.
- **User Privacy:** Regarding the privacy of credential issuance, it's important to note that generating a pre-credential for a claim within the Oracle protocol does not disclose any information about the user. Furthermore, given the commitment's hiding property and the privacy guarantees provided by the Secure Multi-Party Computation (SMPC) evaluation, there is no opportunity for an attacker to gain knowledge about the user during the issuance process. In addition, since no personal information is used when evaluating interaction within BML, there is no risk of de-anonymization or leakage of information about interacting parties.
- **Reputation Binding:** Our DARS is based on the forgery-proof nature of the cryptographic signatures used to create contextual credentials and submit access

tokens. This ensures that the reputation score remains cryptographically linked to the original user or entity.

- **Forward Reputation Binding:** DARS satisfies this property if the signature scheme prevents forgery, the commitment scheme maintains the hiding and binding properties, and the zkSNARK scheme ensures soundness and ZK properties. These combined properties help ensure that the new reputation score is reliable, private (if not shown) and consistently linked to the entity (access token) it represents.

8 Conclusion

In this work, we have proposed a decentralized anonymous reputation system that combines Blockchain and zkSNARKs. The system is built on top of two separate ledgers to decouple identity management from business activities. We make use of Decentralized Oracle Networks not only to automate SCs execution as it is traditionally used but also to import credentials from external systems to prevent sybil attacks without compromising user privacy. DARS users can generate/use as many pseudonyms as they wish on the BC to protect their digital identity and guarantee continued anonymity. The proposed framework relies on two Collision-Resistant Hash-based Merkle trees, UCTree and RCTree, over a list of access and reputation commitments, respectively, to guarantee anonymity while maintaining effective reputation management. We also designed a general reputation model that achieves the security and privacy properties of our formal model. Our design is suitable for all trust-based applications, such as decentralized marketplaces and crowdsourcing platforms. We implemented and tested a prototype of the proposed framework using cutting-edge technologies and cryptographic tools. The results of this evaluation demonstrate the feasibility and effectiveness of DARS. Thus, achieving its objective for practical and realistic usage.
Ongoing research in the fusion of ZKPs and BC shows promising potential. Specifically, advancements in L2 scaling solutions like zkRollups[7] offer hope for substantial improvements. Consequently, our upcoming focus will center on improving the scalability of DARS.

Acknowledgements This work received support from the Nouvelle Aquitaine region through funding from the B4IoT project.

References

1. P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman (2000). Reputation systems. Communications of the ACM 43(12), 45–48.

[7] https://docs.zksync.io/userdocs/intro/

2. Hasan, O., Brunie, L., and Bertino, E. (2022). Privacy-Preserving Reputation Systems Based on Blockchain and Other Cryptographic Building Blocks: A Survey. ACM Computing Surveys (CSUR), 55(2), 1-37.
3. Zheng, Y., Duan, H., and Wang, C. (2018). Learning the Truth Privately and Confidently: Encrypted Confidence-Aware Truth Discovery in Mobile Crowdsensing. IEEE Transactions on Information Forensics and Security, 13(10), 2475-2489.
4. Zhang, Y., Deng, R.H., Zheng, D., Li, J., Wu, P., and Cao, J. (2019). Efficient and Robust Certificateless Signature for Data Crowdsensing in Cloud-Assisted Industrial IoT. IEEE Transactions on Industrial Informatics, 15(9), 5099-5108.
5. M. A. Bouchiha, Y. Ghamri-Doudane, M. Rabah and R. Champagnat (2023). GuRuChain: Guarantee and Reputation-based Blockchain Service Trading Platform. IFIP Networking Conference (IFIP Networking), Barcelona, Spain, 1-9.
6. S. Malik, N. Gupta, V. Dedeoglu, S. Kanhere and R. Jurdak (2021) TradeChain: Decoupling Traceability and Identity in Blockchain enabled Supply Chains. IEEE 20th (TrustCom), Shenyang, China. 1141-1152.
7. Maram, D., Malvai, H., Zhang, F., Jean-Louis, N., Frolov, A., Kell, T., Lobban, T., Moy, C., Juels, A., and Miller, A. (2021). CanDID: Can-do decentralized identity with legacy compatibility, sybil-resistance, and accountability. IEEE Symposium on Security and Privacy (SP 2021). San Francisco, CA, USA. 1348-1366.
8. Abe, M., Fehr, S. (2007). Perfect NIZK with Adaptive Soundness. In: Vadhan, S.P. (eds) Theory of Cryptography. TCC 2007. LNCS vol 4392. Springer, Berlin, Heidelberg. 118-136.
9. D. Benarroch and M. Campanelli and D. Fiore and K. Gurkan and D. Kolonelos (2023). Zero-Knowledge Proofs for Set Membership: Efficient, Succinct, Modular. Designs, Codes and Cryptography, 91, 3457–3525.
10. D. Liu, A. Alahmadi, J. Ni, X. Lin, and X. Shen. (2019). Anonymous reputation system for IIoT-enabled retail marketing atop PoS blockchain. IEEE Transactions on Industrial Informatics 15(6), 3527–3537.
11. K. Zhao, S. Tang, B. Zhao, and Y. Wu. (2019) Dynamic and privacy-preserving reputation management for blockchain-based mobile crowdsensing. IEEE Access 7, 74694–74710.
12. Jo, H.J., Choi, W. (2019) BPRF: Blockchain-based privacy-preserving reputation framework for participatory sensing systems. PLOS ONE 14(12): e0225688.
13. A. Schaub, R. Bazin, O. Hasan, and L. Brunie, (2016). A trustless privacy-preserving reputation system. 31st IFIP Int. Information Security and Privacy Conf. (SEC). Belgium. 398-411
14. K. Soska, A. Kwon, N. Christin, and S. Devadas, (2016). Beaver: A decentralized anonymous marketplace with secure reputation. Cryptology ePrint Archive, 2016/464.
15. Tassos Dimitriou. (2021). Decentralized Reputation. 11th ACM Conference on Data and Application Security and Privacy (CODASPY '21). New York, NY, USA, 119–130.
16. E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, (2014). Zerocash: Decentralized anonymous payments from Bitcoin. IEEE Symp. on Security and Privacy, Berkeley, CA, USA. 459–474.
17. F. Zhang, D. Maram, H. Malvai, S. Goldfeder, and A. Juels, (2020). DECO: Liberating Web Data Using Decentralized Oracles for TLS. ACM SIGSAC Conf. on Computer and Communications Security, USA. 1919–1938.
18. L. Breidenbach, C. Cachin, B. Chan, A. Coventry, S. Ellis, A. Juels F. Koushanfar, A. Miller, B. Magauran, D. Moroz S. Nazarov, A. Topliceanu, F. Tramer, and F. Zhang. (2021) Chainlink 2.0: Next Steps in the Evolution of Decentralized Oracle Networks.
19. Gennaro, R. (2004). Multi-trapdoor Commitments and Their Applications to Proofs of Knowledge Secure Under Concurrent Man-in-the-Middle Attacks. In Advances in Cryptology - CRYPTO 2004. LNCS vol 3152. Springer, Berlin, Heidelberg. 220-236.
20. Groth, J. (2016). On the Size of Pairing-Based Non-interactive Arguments. In Advances in Cryptology – EUROCRYPT 2016. LNCS vol 9666. Springer, Berlin, Heidelberg. 305-326.
21. G. Ariel, Z.J. Williamson, and O. Ciobotaru, (2019). PlonK: Permutations over Lagrange-bases for Oecumenical Noninteractive arguments of Knowledge. Cryptology ePrint Archive, 2019/953.