energies

MDPI

Article

Privacy-Preserving Charging Coordination Scheme for Smart Power Grids Using a Blockchain

Hany Habbak [1], Mohamed Baza [2], Mohamed M. E. A. Mahmoud [3,*], Khaled Metwally [1], Ahmed Mattar [1] and Gouda I. Salama [1]

[1] Department of Computer Engineering and AI, Military Technical College, Cairo 11766, Egypt
[2] Department of Computer Science, College of Charleston, Charleston, SC 29424, USA
[3] Department of Electrical and Computer Engineering, Tennessee Tech University, Cookeville, TN 38505, USA
* Correspondence: mmahmoud@tntech.edu

Abstract: With the rapid emergence of smart grids, charging coordination is considered the intrinsic actor that merges energy storage units (*ESUs*) into the grid in addition to its substantial role in boosting the resiliency and efficiency of the grid. However, it suffers from several challenges beginning with dependency on the energy service provider (*ESP*) as a single entity to manage the charging process, which makes the grid susceptible to several types of attacks such as a single point of failure or a denial-of-service attack (*DoS*). In addition, to schedule charging, the *ESUs* should submit charging requests including time to complete charging (*TCC*) and battery state of charge (*SoC*), which may disclose serious information relevant to the consumers. The analysis of this data could reveal the daily activities of those consumers. In this paper, we propose a privacy-preservation charging coordination scheme using a blockchain. The blockchain achieves decentralization and transparency to defeat the security issues related to centralized architectures. The privacy preservation will be fulfilled using a verifiable aggregation mechanism integrated with an aggregated signing technique to identify the untrusted aggregator and assure the data source and the identity of the sender. Security and performance evaluations are performed, including off-chain and on-chain experiments and simulations, to assess the security and efficiency of the scheme.

Keywords: electrical vehicle; privacy preservation; blockchain; charging coordination; security; smart contract; energy storage units

Citation: Habbak, H.; Baza, M.; Mahmoud, M.M.E.A.; Metwally, K.; Mattar, A.; Salama, G.I. Privacy-Preserving Charging Coordination Scheme for Smart Power Grids Using a Blockchain. *Energies* **2022**, *15*, 8996. https://doi.org/10.3390/en15238996

Academic Editor: J. C. Hernandez

Received: 4 November 2022
Accepted: 25 November 2022
Published: 28 November 2022

1. Introduction

Recently, smart grids have received much attention in industrial, academic, and technological societies. They are considered the smart surrogate for aging power grids. They have considerable potential to provide smart services since they amalgamate several technologies such as the internet of things (*IoT*), big data, cloud computing, etc [1]. The energy storage units (*ESUs*) are the prime actors in a smart grid, which could be a home battery or electric vehicle (*EV*) [2]. They can store energy in case of energy overflows and supply it to the grid in case of difficulties equalizing the energy demand and supply, which can boost the smart grid's endurance [3]. In addition, *ESUs* contribute to the growing popularity of green energy and sustainable development by hoarding the energy overcapacity from renewable energy generators [4]. This hoarded energy could be supplied to powerhouses and charge *EVs* during high-demand intervals, which improves grid reliability. Furthermore, *ESUs* introduce an economical advantage in that they allow consumers to avoid purchasing electricity during high-tariff periods, which, in turn, reduces electricity bills [5].

In spite of their advantages, there are considerable challenges that could affect the efficient integration of *ESUs* into the power grid. Particularly, the concurrent tsunami of unscheduled charging requests could cause an imbalance between the incoming charging requests and the supplied energy, which can produce a thorough failure of the smart

grid [6]. For instance, most *EV* owners charge their vehicles after returning to their homes. To address these impacts, there is an essential demand for a charging coordination technique to avoid the collapse of the energy distribution system and avoid mass blackouts [7].

Although several studies have developed charging coordination techniques [8,9], they suffer from several limitations. Firstly, their centralized architectures make them vulnerable to a single point of failure or denial-of-service (*DoS*) attack. Secondly, several techniques assume that the *ESP* is fully trusted and schedules charging requests fairly. Thirdly, the existing charging coordination techniques normally require the *ESUs* to send data to the *ESP*; these data are used to determine the charging priority of the *ESUs*, and by analyzing the data provided by the *ESUs*, sensitive information may be disclosed such as the locations of the *EVs* and their driving distances and the daily activities of the houses' inhabitants [10]. The analysis of this information enables the attacker to obtain a lot of details related to the client's life patterns such as their work address, health condition, income level, etc. [11].

Motivated by the aforementioned limitations of the literature, in this paper, we propose a privacy-preserving charging coordination scheme using a blockchain. With a blockchain in place, the system can be implemented in a decentralized and transparent fashion, which tackles the previously mentioned security issues relevant to the present centralized approaches [2,12]. The use of a blockchain decentralizes the charging coordination technique making it robust against a single point of failure and other attacks that threaten the availability of the system [13]. However, a blockchain does not provide privacy, and therefore we introduce an aggregated masking scheme as a feasible solution for balancing data utilization and privacy preservation in smart grids. Specifically, *ESUs* report individual masked charging requests including *SoCs* and *TCCs* alongside individual signatures periodically to a validator (local aggregator). In turn, the validator blindly aggregates the incoming charging requests to compute the aggregated charging request CR_{agg}. In order to prevent the local aggregator from returning an invalid result, we use a verification method conducted by another validator (verifier) to ensure the integrity of the aggregation process [14]. In addition, an efficient authentication mechanism is necessary to prove the trustworthiness of the data sources and check the identity of the data sender [15]. Then, the local aggregator broadcasts CR_{agg} so that each *ESU* can calculate the charging schedule locally. If the total charging demand exceeds K_{max}, a subset of *ESUs* with high priority charge without exceeding the available energy for charging.

The remainder of this paper is organized as follows. Section 2 presents the related works. Section 3 introduces the network model, threat model, and design goals. The preliminaries and necessary background information are provided in Section 4. The proposed scheme is demonstrated in Section 5. The security and performance evaluations are discussed in Sections 6 and 7. Finally, the conclusions are given in Section 8.

2. Related Works

In this section, we survey some relevant works in two main parts. First, we discuss the problem of charging coordination in the smart grid. Subsequently, we discuss several works on privacy-preservation schemes in the smart grid.

Charging coordination in smart grids has gained a lot of interest recently. Ota et al. [16] introduced a distributed vehicle-to-grid (*DV2G*) control model to coordinate the charging of *EVs*, whereas in [17], a smart charging method for the charging coordination of plug-in hybrid electric vehicles (*PHEVs*) was introduced. This method intended to reduce the everyday overall charging fees by combining the grid-to-vehicle (*G2V*) and the vehicle-to-grid (*V2G*) methods using real-time tariffs (*RTTs*) in parking lots, which were managed by different aggregators. This enhancement method was merged with the smart charging scheduling algorithm (*SCSA*) to determine the appropriate charging fees and times. In addition, in [18], a new charging scheduling mechanism was proposed based on a double-purpose improvement algorithm to enhance efficiency and reduce fees. The authors

applied other charging solutions such as the vehicle-to-vehicle ($V2V$) and charging-station-to-vehicle ($CS2V$) methods.

In [19], an energy management mechanism was proposed to motivate EV owners to participate in the energy trading process through a game-theoretical scheme. The same topic was introduced in [20], where an energy optimization and auxiliary service scheduling mechanism was proposed to maximize the gain of *EVs*, introduce more resilience, minimize the peak load to the *ESP*, and lower the costs for consumers.

Kang et al. [21] developed a consortium blockchain-based decentralized electricity trading system to charge $V2V$. They applied an optimization mechanism named iterative double auction to improve the charging costs, as well as the amount of commercial power exchanged between *EVs*. In addition, in [22], another decentralized charging coordination mechanism for *EVs* was proposed, which formulated charging coordination for *EVs* as an optimal control problem using the flexibility of EV loads to charge during power consumption valley periods.

In [23], a real-time charging station selection mechanism using large-scale *GPS* data mining was introduced. The mechanism aimed to select the most convenient charging station for *EVs* using their historical charging events and real-time *GPS* data streams. In addition, Cao et al. [24] proposed a regular updating technique for choosing the appropriate charging station in order to solve the problem of *EVs* not reaching the planned charging stations on time. Xu et al. [25] introduced a mathematical model of the ideal charging plan using an analysis of the previous charging behaviors of *EVs*.

Several papers in the literature have investigated security and privacy preservation in smart grids. Akula et al. [26] introduced a secure approach to smart grid power injection. In the proposed approach, the incoming masked bids sent by *ESUs* are aggregated and submitted to the *ESP*. The objective of this work was to enable the *ESP* to learn the total amount of power that can be injected by the *ESUs* in a confidential method. In addition, several reliable aggregation schemes based on secure matrix multiplications over encrypted data were proposed in [27]. These schemes applied the k-nearest neighbor (kNN) similarity mechanism.

In [28], a data obfuscation method was used in an AMI network to develop protected and operative mechanisms to distribute obfuscated data. Li et al. [29,30] introduced an anonymous and authenticated mechanism to preserve the privacy of the stored energy reported from *EVs*. Then, they developed an optimal authentication method to secure the location of the *EVs*.

In [31], a consortium blockchain-based data aggregation and regulation scheme for smart grids was introduced. This scheme is concerned with multidimensional data collection and multi-recipients in the consortium blockchain. In [32], a blockchain and fog-computing-based privacy-preserving charging mechanism for *EVs* was proposed. Fog computing was used to reduce the overload on the server side and introduce local computing services. In addition, the blockchain system was deployed on the distributed fog-computing nodes (*FCNs*), introducing a decentralized and secure storage media. The security of the communication between *EVs* and *FCNs* was achieved using a mutual authentication scheme.

In [9], two privacy-preserving and collusion-resistant charging coordination approaches for smart grids were discussed. The first was a centralized approach in which *ESUs* authenticated their charging requests anonymously using anonymous and unlinkable tokens obtained from a centralized server operated by the electrical utility. Moreover, *ESUs* sent multiple charging requests with random *TCC* and *SoC* data in a truncated normal distribution to prevent linking the charging requests sent by an ESU to preserve privacy. Furthermore, in the decentralized approach, secret masks are shared among proxy *ESUs* to thwart collusion attacks. Wang et al. [33] introduced an identity-based verifiable aggregator's oblivious encryption scheme for smart grids. They proved the aggregator obliviousness and unforgeability through the smooth projective hash function and computational Diffie–Hellman presumption.

In addition, they developed an identity-based aggregation protocol for smart grids based on their proposed encryption scheme.

In [34], a lightweight data aggregation scheme for smart grids was proposed. The scheme is suitable for devices with limited resources, such as smart meters and task schedulers that perform only lightweight computations, whereas complex operations are performed by scalable processing units. Another lightweight privacy-preserving data aggregation scheme for smart grids was proposed in [35]. The scheme is suitable for devices with limited resources and thwarts collusion attacks for up to $(n-1)$ users.

Although several studies have developed charging coordination techniques [8,9], they suffer from several limitations. Firstly, the current techniques often have a centralized architecture that makes them vulnerable to a single point of failure, i.e., they depend on a single server to manage and coordinate incoming charging requests. If a successful denial-of-service (*DoS*) attack is launched on the energy service provider (*ESP*), the entire system fails. Secondly, the existing techniques assume that the *ESP* is fully trusted and schedules charging requests fairly. Specifically, there is no guarantee that the charging coordination technique is executed precisely. Thirdly, the existing charging coordination techniques normally require the *ESUs* to send data to the *ESP* such as time to complete charging (*TCC*) and battery state of charge (*SoC*). In particular, these data are used to determine the charging priority of the *ESUs*, and then the *ESUs* with the highest priority charge first without exceeding the maximum available power (K_{max}), whereas the charging of the other *ESUs* is deferred to the next time intervals. Therefore, by analyzing the data provided by *ESUs*, sensitive information may be disclosed such as the locations of the *EVs* and their driving distances and the daily activities of the houses' inhabitants [10]. The analysis of this information enables the attacker to obtain a lot of details related to the client's life patterns such as their work address, health condition, income level, etc. [11]. In some cases, this information could be sold or exchanged for commercial purposes. In other cases, the charging requests sent by *EVs* could determine whether an *EV* owner is at home, how long he/she will stay there, and how frequently he/she drives. For instance, if a home battery is not charged for an extended period, this indicates that the residents are not at home or are traveling and, consequently, this house could be broken into [36].

3. Network/Threat Models and Design Goals

In this section, we first introduce the network model of our proposed scheme, followed by the threat model, and finally, the design goals of the scheme.

3.1. Network Model

As illustrated in Figure 1, the network model of our scheme involves three main entities: the key management center (*KMC*), energy storage units (*ESUs*), and blockchain network.

- **The Key Management Center (KMC):** The *KMC* is a trusted party. It can be a governmental authority such as the Ministry of Electricity and Energy, which is the administrator of the whole system. It is responsible for the initialization of the entire system including the registration of all terminals.
- **Energy Storage Units (ESUs):** An *ESU* can be either a home battery or EV. It is the fundamental unit in the smart grid, which is deployed in one community named group and connected to the blockchain network through the validator (local aggregator).
- **The Blockchain Network:** The blockchain is the heart of our proposed scheme. It is responsible for receiving and scheduling incoming charging requests without exceeding the maximum energy capacity or accessing individual charging requests. The blockchain consists of nodes named validators. Each validator acts as either an aggregator or a verifier. The local aggregator receives n masked signed charging requests from $nESUs$ in one group. It computes the aggregated charging request (CR_{agg}). Finally, the aggregation results and proof of correct aggregation are forwarded to a verifier to start the verification process. The verifier uses the reported verification parameters to check the correctness of the aggregated result. In addition, due to

the nature of the blockchain, it is a common assumption that more than 50% of the blockchain validators are honest. In this paper, we implement our proposed scheme using a private blockchain. A private blockchain is selected because each verifier belongs to a different utility and every utility is considered a node in the blockchain network. In addition, a private blockchain provides higher throughput with more acceptable latency than a public blockchain.

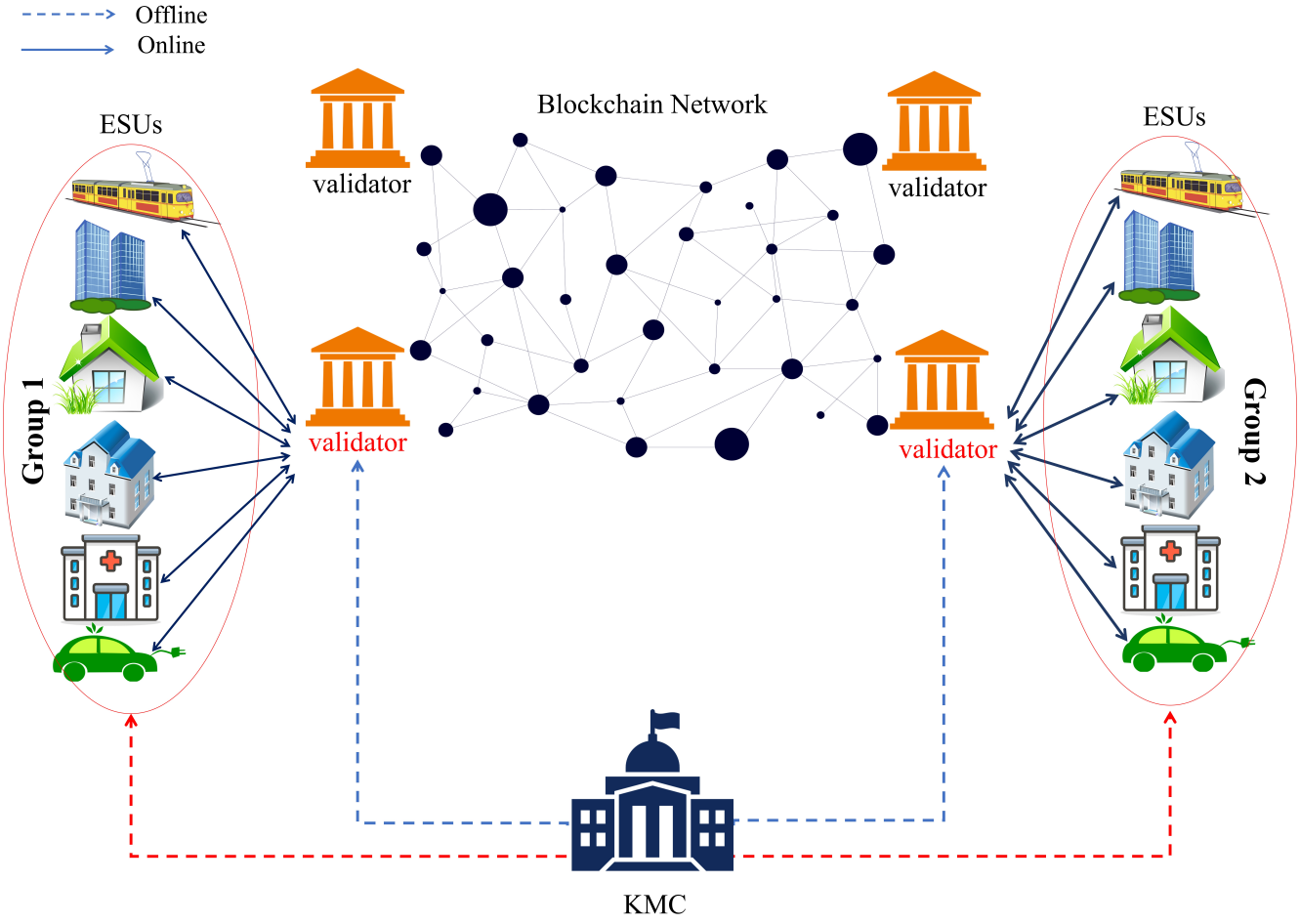


Figure 1. An illustration of the network model.

3.2. Threat Model

The *KMC* is a trusted party since it is supervised by a governmental authority that is concerned with the security of the entire system. The blockchain network is a conceptual trusted party that is designed for data immutability and transparency but not for privacy preservation [37]. The blockchain validators could misbehave with the aggregated data or try to infer sensitive information. In this paper, we assume that blockchain validators (local aggregators) are curious to obtain information about *ESUs*' owners. In addition, some of these validators may be malicious and they do not perform the computations honestly. Moreover, the attackers may passively snoop on the communications between *ESUs* and the local aggregator to infer sensitive information such as whether inhabitants of a house are currently traveling, their return home time, and other daily activities.

3.3. Design Goals

We demonstrate that our proposed scheme fulfills the following substantial objectives:

- **Decentralization and resiliency:** Our proposed scheme should not rely on any central party to perform the charging coordination. As mentioned previously, centralized

schemes are vulnerable to single-point-of-failure and other attacks and suffer from a lack of transparency. Therefore, our proposed scheme considers that no entity in the entire system has complete central authority to control the charging coordination process.

- **Privacy-preserving charging scheduling:** Our proposed charging scheduling scheme should compute the charging schedules without exceeding the K_{max} and without disclosing any sensitive information about the clients.
- **Resistance to replay attacks:** Our proposed scheme should counter replay attacks, i.e., if an attacker tries to record a charging request and sends it later, it should be rejected.
- **Data integrity and authenticity:** Our proposed scheme should guarantee the integrity and authenticity of incoming charging requests and also the authenticity of the *ESUs*' identity.

4. Preliminaries

In this section, we introduce the essential background for this paper including bilinear pairing, the computational Diffie–Hellman problem, the key agreement protocol, the deterministic random generator, and the blockchain. All notations used in this paper are presented in Table 1.

Table 1. Notations.

Notation	Description
ESU_i	Energy Storage Unit
ESP	Energy Service Provider
$\mathbb{P}_i$	ESU_i priority
$\mathbb{S}_i$	ESU_i SoC
$\mathbb{T}_i$	ESU_i TCC
$\{G_1, G_2, G_T, g_1, g_2, \hat{e}, p, H_1, H_2, \lambda, \Upsilon\}$	Public Parameters
$\langle(\lambda, \Upsilon), \eta\rangle$	$KMC's$ master public values, master secret key
δ	Common State Value
ID_i	The Identity of ESU_i
K_{pubi}, K_{privi}	ESU_i Public Key and Private Key Pair
$\langle\chi_i, \Lambda_i\rangle$	ESU_i Private Key
CR_i	ESU_i Charging Request
CR_{agg}	Aggregated Charging Request
γ_1,γ_2	The Equation Weights
L_m	Priority Label
Ts	Temporal Charging Slot
π	$24/Ts$
R_i	Proof Value
PV_i	$ESU_i's$ Public Value
Γ_i	$ESU_i's$ Masked Charging Request
u_{ij}	Shared Seed Value Between ESU_i and ESU_j
$\langle W_i, N_{i1}, N_{i2}, C\rangle$	$ESU_i's$ Signature Credentials
$\sigma_i = (V_i, W_i)$	$ESU_i's$ Signature
ρ	Correctness Proof of The Aggregated Value
$\mathbf{V}$	Aggregated V
$\mathbf{W}$	Aggregated W
$\boldsymbol{\sigma}$	Aggregated σ
K_{max}	Maximum Available Power
POW_{Lm}	The Power Assigned to The ESU In Terms of Priority Labels

4.1. Bilinear Pairings

Let $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are bilinear groups with prime order p, in which g_1, g_2 are the generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively.

Definition 1. *A pairing* $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$ *has the following properties:*

- *Non-degeneracy:* $\hat{e}(g_1, g_1') \neq 1$.
- *Bilinearity:* $\hat{e}(\nu g_1, \iota g_1') = \hat{e}(g_1, g_1')^{\nu\iota}$ $\forall g_1, g_1' \in \mathbb{G}_1$ *and* $\nu, \iota \in \mathbb{Z}_p^*$ *where* $\mathbb{Z}_p^*$ *is a finite field of order* p.
- *Computability: There is an efficient algorithm to compute* $\hat{e}(g_1, g_1')\forall g_1, g_1' \in \mathbb{G}_1$.

Definition 2. *A pairing* $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ *has the following properties:*

- *Non-degeneracy:* $\hat{e}(g_1, g_2) \neq 1$.
- *Bilinearity:* $\hat{e}(g_1^{\nu}, g_2^{\iota}) = \hat{e}(g_1, g_2)^{\nu\iota}\forall g_1 \in \mathbb{G}_1, g_2 \in \mathbb{G}_2$ *and* $\nu, \iota \in \mathbb{Z}_p^*$ *where* $\mathbb{Z}_p^*$ *is a finite field of order* p.
- *Computability: There is an efficient algorithm to compute* $\hat{e}(g_1, g_2)\forall g_1; g_2 \in \mathbb{G}_1, \mathbb{G}_2$.

4.2. Computational Diffie–Hellman (CDH) Problem

Let $\mathbb{G}_1$ be a cycle additive group of prime order p, g_1 is a generator of $\mathbb{G}_1$, $\forall \nu, \iota \in \mathbb{Z}_p^*$, given $\langle g_1, \nu g_1, \iota g_1 \rangle$, compute $\langle \nu\iota g_1 \rangle \in \mathbb{G}_1$. It is assumed that the CDH problem is very hard.

4.3. Key Agreement Protocol

A key agreement protocol (KAP) involves a set of algorithms as follows:

- $KAP.ParmGen : (\varepsilon) \rightarrow (PubParm)$. The algorithm takes the security parameter ε as input and produces public parameters $(\mathbb{G}_1, g_1, p, H_1)$, where $\mathbb{G}_1$ is a group with a prime order p, where $p > 2^{\varepsilon}$, and H_1 is a hash function.
- $KAP.KeyGen : (PubParm) \rightarrow (K_{privi}, K_{pubi})$ using public parameters, the terminal i could generate its private\public key pair $KAP.KeyGen(PubParm) \rightarrow (\eta_i, g_1^{\eta_i})$, in which $\eta_i \in_r \mathbb{Z}_p^*$ represents the private key of terminal i and $g_1^{\eta_i}$ is the corresponding public key.
- $KAP.KeyAgree : (K_{pubi}, K_{privi}) \rightarrow (u_{ij})$. Using the private key of terminal i, the public key of terminal j, and the $KAP.KeyAgree$ algorithm, such as the Diffie–Hellman key agreement protocol, the two terminals could generate a shared key u_{ij}.

4.4. The Deterministic Random Generator (DRG)

The DRG is an algorithm for generating a sequence of numbers, whose characteristics are roughly similar to the characteristics of sequences of real random numbers. This generated sequence is not purely random since it is completely determined by an initial value called a seed. In our proposed scheme, the DRG is loaded with an identical random seed of some constant length for every two units. It ensures that the generated value based on the random seed is computationally indistinguishable from an identical sampled element from the output space as long as the attackers can not compute this seed. In our proposed scheme, the DRG plays the main role by generating random numbers that could be used in the masking operation.

4.5. Blockchain Technology

As illustrated in Figure 2, the blockchain manages a decentralized, verifiable, immutable, and distributed ledger that permits distrusted parties to transact securely and agree on a unified shared ledger without the need for a central entity [38]. It introduces impartiality, probity, and integrity, where the data of the shared ledger are arranged as a chain of blocks and administrated by a network of computers/servers and named miners/validators operating a peer-to-peer (P2P) protocol. Every block contains a set of transactions carried out by the network peers and is verified by all the network nodes according to a predetermined agreement mechanism (consensus algorithm) [39]. This consensus and other stimulant techniques of the blockchain help to support and fortify the reciprocal confidence among the network nodes. Typically, a blockchain has two main types [40], private and public. In private (permissioned) blockchains, access to reading/writing is allowed for

authorized users but in public (permissionless) blockchains, access to reading/writing is permitted for anyone, and all users have the same reading/writing privileges. In this paper, we implement our proposed scheme using a private blockchain. A private blockchain is selected because each verifier belongs to a different electrical utility and every utility acts as a node in the blockchain network. In addition, a private blockchain provides a higher throughput with more acceptable latencey than a public blockchain.

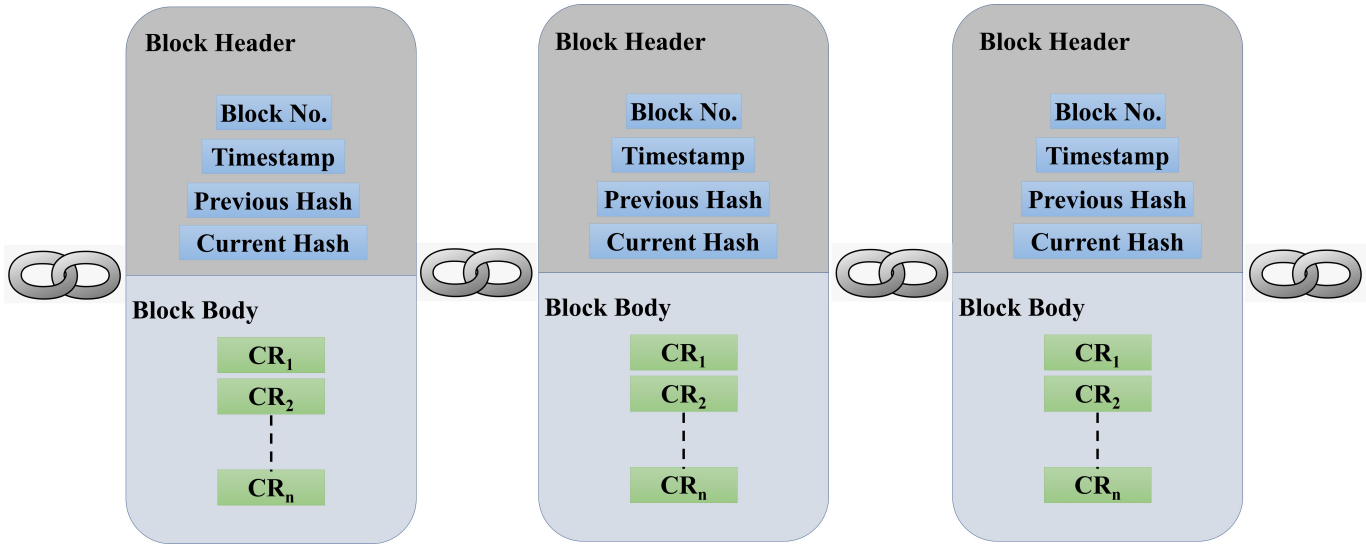


Figure 2. The structure of the charging request ledger.

5. The Proposed Scheme

This section provides a depiction of our proposed privacy-preserving charging coordination scheme using a blockchain. It consists of four sequential phases that begin with the system initialization phase, which occurs only once when the scheme is first deployed. It involves the generation of all cryptographic credentials and public parameters for the entire system. This phase is conducted by the *KMC*. Then, there is the submission of the charging request phase, which is conducted by the *ESUs* and includes two main sub-phases, preparing the charging requests and submitting them. This phase involves running the *KAP* by the *ESUs* in the same community to compute the necessary seed to start the masking process, masking charging requests using a one-time mask technique, and individual signing operation, and then submitting them to the local aggregator. In addition, it creates the generic verification parameter (*GVP*) that is used by the verifier(s) to verify the correctness of the aggregation. Next, is the aggregation and verification of the charging requests phase, where the local aggregator aggregates the individually signed masked charging requests to produce the CR_{agg}. Thereafter, it checks the aggregated signature and computes the proof of the aggregation process that is sent to the verifier(s) to prove the integrity of the aggregation process. Finally, in the computing charging schedules phase, once the verifier checks the proof, the local aggregator broadcasts the CR_{agg} to all of the *ESUs*. Each *ESU* locally computes its charging schedule to determine whether it could charge in the present time slot or it has to send its charging request to the next time interval. All the exchanged messages in our proposed scheme are shown in Figure 3.

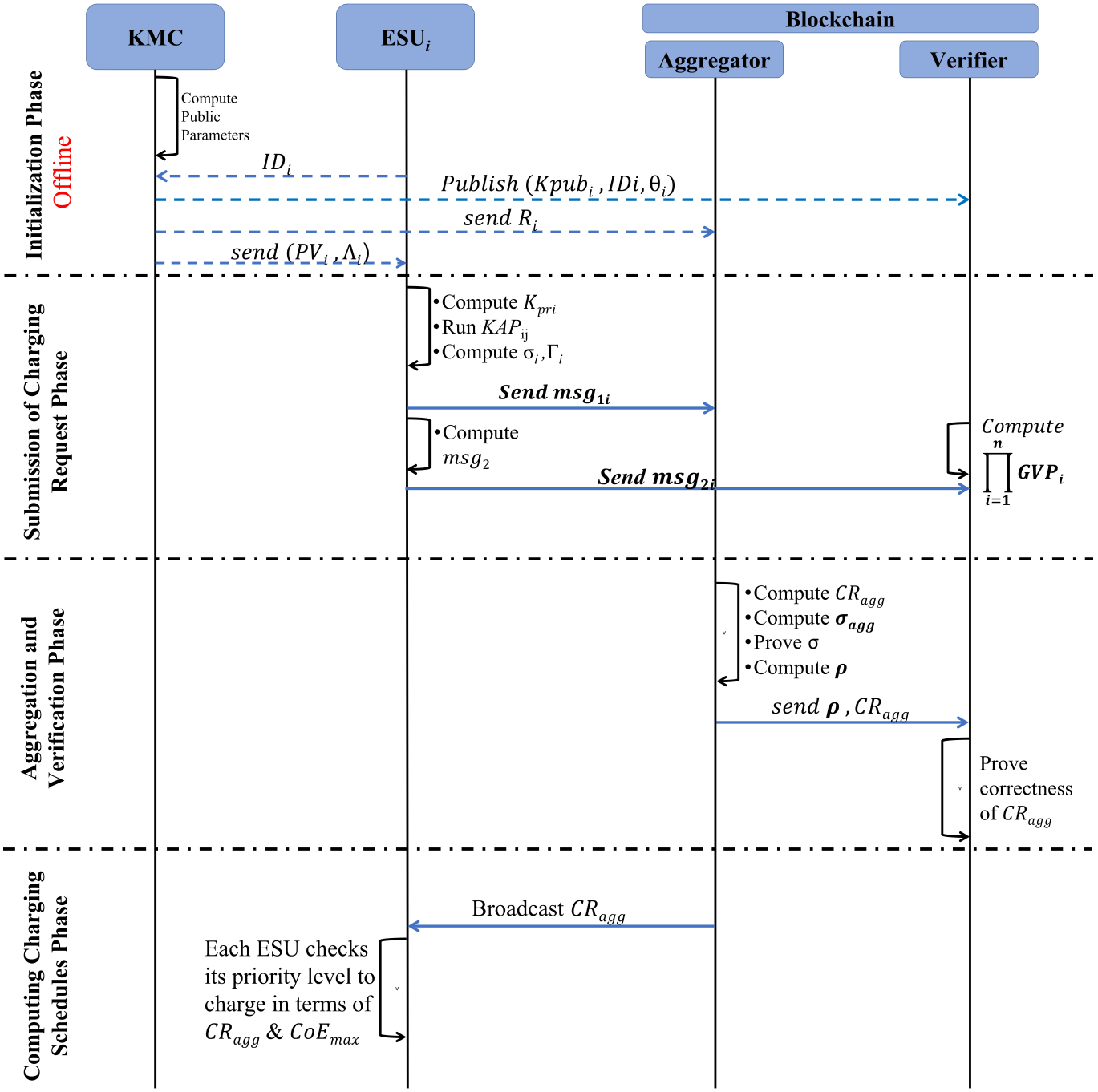


Figure 3. An integrated illustration of the proposed scheme.

5.1. System Initialization

In this phase, the *KMC* takes the security parameter ε as the input and then selects the bilinear group parameters. $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are bilinear groups with prime order p, where $p > 2^{\varepsilon}$. g_1 and g_2 are the generators of G_1 and G_2, respectively. Two cryptographic hash functions H_1 and H_2 are selected by the *KMC*, where $H_1 : \{0,1\}^* \longrightarrow G_1$ and $H_2 : \{0,1\}^* \longrightarrow \mathbb{Z}_p^*$. Then, the *KMC* chooses $\eta \in \mathbb{Z}_p^*$ where η is the master secret key of the *KMC* and calculates two master public values $\lambda = \eta g_1$ and $\Upsilon = g_2^{\eta}$. The *KMC* publishes the following parameters $Param = \{G_1, G_2, G_T, g_1, g_2, \hat{e}, p, H_1, H_2, \lambda, \Upsilon\}$. The common state value (δ) is a stochastic size string picked up randomly by the *KMC*. It could be a date or any part of a common parameter [15,41].

ESU_i sends a registration request to the KMC by sending its ID_i; the KMC verifies this ID_i and then computes $\theta_i = H_1(ID_i)$. It also computes $\Lambda_i = \eta\theta_i$ as a part of the private key of the ESU_i. The ESU_i chooses a random value $\chi_i \in_R \mathbb{Z}_p^*$ and then it computes its corresponding public key $K_{pubi} = \chi_i g_1$ and also the corresponding private key $K_{privi} = \langle \chi_i, \Lambda_i \rangle$. Eventually, KMC publishes $(K_{pubi}, ID_i, \theta_i)$.

The KMC uses Λ_i to compute R_i, which is used as a proof by the local aggregator for correct aggregation, where $R_i = g_1^{(\eta+\Lambda_i)}$, and creates a public value $PV_i = \hat{e}(g_1^{\Lambda_i}, g_2)$. The KMC sends R_i to the local aggregator and sends PV_i to the ESU_i.

5.2. Submission of Charging Requests

This phase consists of two main sub-phases, preparing a charging request and submitting it.

5.2.1. Preparing Charging Requests

All *ESUs* send their charging requests to the local aggregator and each *ESU* computes its priority index $\mathbb{P}_i$ using the following equation:

$$\mathbb{P}_i = \gamma_1(1 - \mathbb{S}_i) + \gamma_2\Phi(\mathbb{T}_i) \quad (1)$$

where $\mathbb{P}_i$ is the priority of an ESU_i. *SoC* is denoted by $\mathbb{S}_i \in [0, 1]$, in which if $\mathbb{S}_i = 1$, it indicates that ESU_i is fully charged and $\mathbb{S}_i = 0$ indicates fully uncharged. The priority index increases as the *SoC* decreases (*ESUs* with lower energy have higher priority). $\mathbb{T}_i$ is the $TCC \in [0, 1]$ and $\Phi(\mathbb{T}_i)$ is a decreasing function, where $\Phi(\mathbb{T}_i) = 0$ for a long $\mathbb{T}_i$ and $\Phi(\mathbb{T}_i) = 1$ for a short $\mathbb{T}_i$, i.e, the priority increases as the TCC becomes shorter. γ_1 and γ_2 are weights to provide the relative significance for $\mathbb{S}_i$ and $\mathbb{T}_i$, with $\gamma_1 + \gamma_2 = 1$.

The priority index is categorized into labels, such as $\{L_1, L_2, \ldots L_m\}$, so if we classify them on a scale of 1 to 10, they could be sorted as follows $L_1 \in [0, 0.1], L_2 \in [0.1, 0.2], \ldots L_{10} \in [0.9, 1]$. Every charging request is split into groups of bits, where each group represents a specific priority label and every *ESU* provides its charging demand in the equivalent group to its priority label L_i. For example, if the CR length is 500 bits and we have 10 priority labels, the message is divided into 10 groups, each of them with a length of 50 bits. When an *ESU* submits its CR, it should store its charging demand in the equivalent group of its priority label and null in the other groups. As shown in Figure 4, the request from an ESU_1 named CR_1 and its corresponding priority label is L_1 so if the charging demand of ESU_i is 10 KW, it could be presented as 10 in group 1 (the rightmost 50 bits) and null in the other groups. Therefore, when all *CRs* are aggregated, to avoid arithmetic overflow, each group should be allocated an appropriate number of bits to prevent a carry to the next priority label group from occurring. Finally, the CR_{agg} provides the total charging needs of all *ESUs* and each priority label [9].

5.2.2. Submitting Charging Requests

The day is divided into a number of temporal slots $\{1, 2, ..., \pi\}$ with uniform periods T_s that cover the entire 24 hours of the day, where $\pi = 24/T_s$. ESU_i submits its CR_i during a predefined temporal slot T_s [5].

At the beginning of this phase, ESU_i executes the KAP with each of the other *ESUs* to procure several seed values according to the number of other *ESUs*. To reduce the overhead of computing the seeds, a seed value u_{ij} is inputted to a DRG to generate a random value that is used in the masking process. This technique is used to compute a one-time mask, u_{ij}, which is shared between every pair of *ESUs* i and j, where $(i \neq j)$, and the masks are used to hide the charging requests of the individual *ESUs* to preserve privacy. Each ESU_i adds the masks shared with ESU_j to CR_i $if(i < j)$, otherwise it subtracts it from CR_i as follows:

$$\Gamma_i = CR_i + (\sum_{j=i+1}^{n} DRG(u_{ij}) - \sum_{j=1}^{i-1} DRG(u_{ij})) \quad (2)$$

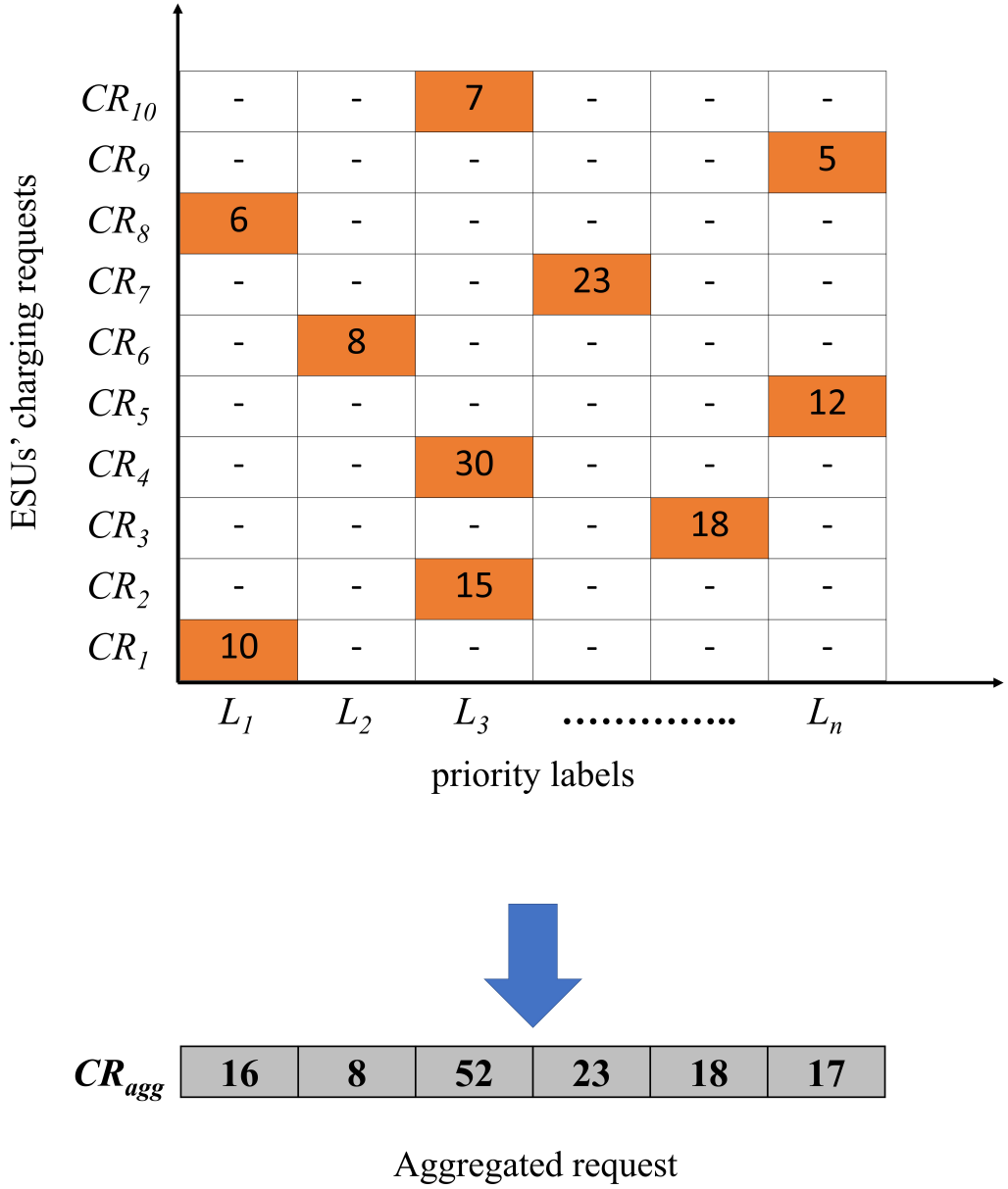


Figure 4. The format of the individual charging requests and the aggregated request in terms of the priority index.

The masking operation is carried out by ESU_i to mask CR_i and to create the generic verification parameter GVP_i that is used by the verifier to verify the integrity of the CR_{agg} [14]. ESU_i uses δ, ID_i and the key pairs of the ESU_i (public key, private key) $(K_{pubi}, \langle \chi_i, \Lambda_i \rangle)$ to sign its CR_i as follows. First, ESU_i chooses a random number $r_i \in_R \mathbb{Z}_p^*$ and then computes

$$W_i = r_i g_1$$
$$N_{i1} = H_2(CR_i \parallel \delta \parallel ID_i)$$
$$N_{i2} = H_2(CR_i \parallel \delta \parallel K_{pubi})$$
$$C = H_1(\delta \parallel \lambda)$$
$$V_i = N_{i2} \cdot \Lambda_i + (\chi_i \cdot N_{i1} + r_i) \cdot C$$

Finally, ESU_i obtains the signature σ_i, where $\sigma_i = (W_i, V_i)$. Then, ESU_i reports its signed masked charging request (msg_1) to the local aggregator including the masked charging request Γ_i, signature σ_i, and timestamp τ as follows:

$$msg_{1i} = \langle \Gamma_i, \sigma_i, \tau \rangle \quad (3)$$

In this verification process, ESU_i submits the second message (msg_2) to the verifier, where msg_2 contains $GVP_i = PV_i^{\Gamma_i}$ to be used by the verifiers to verify the integrity of the aggregation process that is carried out by the local aggregator.

5.3. Aggregation and Verification of Charging Requests

In this phase, there are two steps. The first step is the verifiable aggregation that is conducted by the local aggregator, and the second step is performed by the validators, called verifiers.

In the aggregation step, the aggregator receives n messages that need to be aggregated and verified to begin the next phase. All the charging requests are signed and masked using the common state value δ. The local aggregator first checks the timestamp τ of each message to ensure that it matches the current time slot. The aggregator has a complete list of $\{ID_i\}_{i=1}^n$, $\{\theta_i\}_{i=1}^n$, the corresponding public keys $\{K_{pubi}\}_{i=1}^n$, the equivalent signatures $\sigma_i = (W_i, V_i) \forall (1 \leq i \leq n)$, and the masked charging request $\Gamma_i \forall (1 \leq i \leq n)$. The aggregator computes the aggregated charging request $CR_{agg} = \sum_{i=1}^n \Gamma_i$. In addition, it generates proof of the correctness of the aggregated value. Note that the local aggregator only calculates the CR_{agg} and cannot compute the individual CR of any ESU to preserve privacy. Then, the aggregator verifies the signatures as follows. It first calculates $\mathbf{V} = \sum_{i=1}^n V_i$ and $\mathbf{W} = \sum_{i=1}^n W_i$ to produce the aggregated signature $\sigma = (\mathbf{W}, \mathbf{V})$ for all received CRs, then the aggregator computes the following:

$$\begin{aligned} &C = H_1(\delta \parallel \lambda); \\ &N_{i1} = H_2(CR_i \parallel \delta \parallel ID_i); \\ &N_{i2} = H_2(CR_i \parallel \delta \parallel K_{pubi}) \end{aligned}$$

Then, the aggregator applies the following equation to verify the aggregated signature, and if the equation holds, the signatures are accepted, or else they are rejected.

$$\hat{e}(\mathbf{V}, g_1) \stackrel{?}{=} \hat{e}(\sum_{i=1}^n N_{i2}\theta_i, \lambda)\hat{e}(\sum_{i=1}^n N_{i1}K_{pubi} + \mathbf{W}, C) \tag{4}$$

The proof of this equation is as follows:

$$\begin{aligned} \hat{e}(\mathbf{V}, g_1) &= \hat{e}(\sum_{i=1}^n (N_{i2} \cdot \Lambda_i + (\chi_i \cdot N_{i1} + r_i)C), g_1) \\ &= \hat{e}(\sum_{i=1}^n N_{i2} \cdot \Lambda_i + \sum_{i=1}^n (\chi_i \cdot N_{i1} + r_i)C, g_1) \\ &= \hat{e}(\sum_{i=1}^n N_{i2} \cdot \Lambda_i, g_1)\hat{e}(\sum_{i=1}^n (\chi_i \cdot N_{i1} + r_i)C, g_1) \\ &= \hat{e}(\sum_{i=1}^n N_{i2} \cdot \eta\theta_i, g_1)\hat{e}(\sum_{i=1}^n (\chi_i \cdot N_{i1} + r_i)g_1, C) \\ &= \hat{e}(\sum_{i=1}^n N_{i2} \cdot \theta_i, \lambda)\hat{e}(\sum_{i=1}^n N_{i1}K_{pubi} + \sum_{i=1}^n W_i, C) \\ &= \hat{e}(\sum_{i=1}^n N_{i2}\theta_i, \lambda)\hat{e}(\sum_{i=1}^n N_{i1}K_{pubi} + \mathbf{W}, C) \end{aligned}$$

Once the aggregator computes CR_{agg} and verifies the aggregated signature σ, it computes the proof of the correctness of the aggregation proof ρ. The aggregator sends the generated proof to the verifier to verify the aggregation results.

$$\rho = \prod_{i=1}^n R_i^{\Gamma_i} \forall (1 \leq i \leq n)$$

The verifier receives $\langle CR_{agg} = \sum_{i=1}^{n} \Gamma_i, \{\rho_i\}_{i=1}^{n}, \{GVP_i\}_{i=1}^{n} \rangle$ and executes the verification process to prove the following:

$$\hat{e}(\rho, g_2) \stackrel{?}{=} \hat{e}(g_1^{CR_{agg}}, \Upsilon). \prod_{i=1}^{n} GVP_i \tag{5}$$

and the proof of this equation is as follows:

$$\begin{aligned}
\hat{e}(\rho, g_2) &= \hat{e}(g_1^{\eta \sum_{i=1}^{n} \Gamma_i + \sum_{i=1}^{n} \Lambda_i \Gamma_i}, g_2) \\
&= \hat{e}(g_1, g_2)^{\eta \sum_{i=1}^{n} \Gamma_i} \cdot \hat{e}(g_1, g_2)^{\sum_{i=1}^{n} \Lambda_i \Gamma_i} \\
&= \hat{e}(g_1^{CR_{agg}}, g_2)^{\eta} \cdot \prod_{i=1}^{n} \hat{e}(g_1, g_2)^{\Lambda_i \Gamma_i} \\
&= \hat{e}(g_1^{CR_{agg}}, \Upsilon) \cdot \prod_{i=1}^{n} \hat{e}(g_1^{\Lambda_i}, g_2)^{\Gamma_i} \\
&= \hat{e}(g_1^{CR_{agg}}, \Upsilon) \cdot \prod_{i=1}^{n} GVP_i
\end{aligned}$$

5.4. Computing Charging Schedules

Once the aggregator computes the CR_{agg}, it verifies the signature σ and receives approval from the public verifier, and broadcasts the CR_{agg} to all the *ESUs*. Then, if the total amount of charging demand for all priority labels is less than or equal to the available energy for charging, all charging requests are met because there is enough energy to charge all *ESUs*. Otherwise, a charging coordination technique should be used to select a subset of *ESUs* to charge without exceeding the available energy for charging K_{max}.

Every ESU_i individually performs a contrast operation between the assigned K_{max} and CR_{agg}^{l} according to the priority label L_m (from the maximum to the minimum) until it reaches the threshold priority label that can satisfy $\sum_{l=L_m}^{L_{max}} CR_{agg}^{l} \leq K_{max}$, where L_{max} is the highest priority label.

In the case of $\sum_{l=L_m}^{L_{max}} CR_{agg}^{l} = K_{max}$, all *ESUs* with priority labels greater than or equal to L_m charge in this time interval, and the other *ESUs* need to send charging requests in the next time period. However, if $\sum_{l=L_m}^{L_{max}} CR_{agg}^{l} < K_{max}$, then all *ESUs* with priority labels greater than or equal to L_m charge the amount of energy they demand, and to avoid under-utilizing the available charging energy, the remaining energy is distributed to the *ESUs* with priority labels L_{m-1}, and their charging requests are as follows:

$$POW_{Lm-1} = \Omega \times \left(\frac{CR_{Lm-1}}{CR_{agg}^{L_{m-1}}}\right) \tag{6}$$

where POW_{Lm-1} is the power assigned to the *ESU* with priority label L_{m-1}, CR_{Lm-1} is the original charging demand of an *ESU* with priority label L_{m-1}, $CR_{agg}^{L_m}$ is the aggregated charging demand of the *ESUs* with priority label L_{m-1}, and $\Omega = K_{max} - \sum_{l=L_m}^{L_{max}} CR_{agg}^{l}$.

6. Security Analysis

Our scheme leverages several techniques such as bilinear pairing, verifiable privacy-preserving data aggregation, and verifiable data-aggregated signatures, as demonstrated in Section 4. We assume that these techniques are secure and their security is proved in detail in their papers. Based on this assumption, in this section, we explain how our scheme achieves the security objectives, as explained in Section 3.2.

Proposition 1. *Our proposed scheme preserves the privacy of* ESUs *using a masking-based data aggregation mechanism.*

Proof. In our proposed scheme, neither the local aggregator nor any other *ESU* could know the charging request of any individual *ESU*. This is fulfilled by using a one-time masking-based data aggregation mechanism. Due to using the aggregation mechanism, only the aggregated charging request CR_{agg} could be known in order to compute the charging schedule. In addition, the aggregation mechanism prevents linking charging requests with the identities of the *ESUs*. This is because *ESUs* use different masking values in each request so that two charging requests with the same demands and belonging to the same *ESU* are different. □

Proposition 2. *External eavesdroppers cannot infer any sensitive information about consumers.*

Proof. In our proposed scheme, the *SoC* and *TCC* values are masked with random numbers generated by the *DRG*, and the seed of the *DRG* is a secret value shared between two *ESUs* and is computed using the *KAP* technique. Consequently, external eavesdroppers who intercept the charging requests submitted in our proposed scheme cannot obtain the *SoC* and *TCC* or the charging schedule of any *ESU* because it is infeasible to compute the masked random values shared among the *ESUs*. □

Proposition 3. *Our proposed scheme thwarts collusion attacks.*

Proof. Our proposed scheme can thwart collusion attacks since collusion between *ESUs* does not result in acquiring the *SoC* or *TCC* values of any *ESU* due to the use of the different masking keys shared among the *ESUs*. On the other hand, the nature of the blockchain prevents any kind of collusion between the participants. □

Proposition 4. *Our proposed scheme does not suffer from a single point of failure.*

Proof. Due to the decentralization nature of the blockchain network, our proposed scheme is robust and fortified against a single point of failure, which the centralized schemes suffer from. In addition, our charging coordination mechanism is executed by the *ESUs* in a decentralized way and, consequently, it is robust against the single point of failure. □

Proposition 5. *Our proposed scheme counters attacks targeting availability.*

Proof. Our proposed scheme thwarts *DoS*/*DDoS* attacks. In these attacks, the attacker tries to fully or partially suspend the charging coordination mechanism, e.g., by targeting a server or a central unit in the case of centralized systems. However, in order to successfully execute these attacks in our decentralized charging coordination mechanism, the attacker should target all *ESUs* or the majority of them, which is practically infeasible. □

Proposition 6. *Our scheme thwarts replay attacks.*

Proof. In replay attacks, the attackers record messages submitted by the *ESUs* and replay them later, for example, to impersonate *ESUs* or launch *DoS* attacks. In our proposed scheme, the charging request messages have a common state value and timestamp, which match a specific time interval. Therefore, if the aggregator finds out that one of them is unmatched, it is discarded. □

Proposition 7. *Our scheme counters impersonation, forgery, and data modification attacks.*

Proof. All the messages in our proposed scheme are signed by *ESUs* to guarantee their integrity and prove that they are submitted by authorized clients. Consequently, im-

personation, forgery, and data modification attacks are infeasible since to launch these attacks, the attackers need to know the secret keys of the *ESUs* to be able to compute the valid signatures. □

Proposition 8. *Our proposed scheme is secure against malicious aggregators.*

Proof. In our scheme, malicious aggregators that want to infer sensitive information or report false aggregated values can be identified using a verifiable aggregation technique as follows. First, in our scheme, the aggregator cannot obtain an individual CR and can only compute the CR_{agg}. Second, after the completion of the aggregation, the aggregator generates proof to ensure the correctness of the aggregation process and sends it to a validator (verifier) through the blockchain network. At the same time, each ESU sends a GVP_i to use in the verification process. Thus, it is practically impossible that the aggregator can infer any information relevant to the individual CRs or manipulate the aggregation results. □

Proposition 9. *Our proposed scheme guarantees the integrity and transparency of the charging scheduling process.*

Proof. Our scheme can ensure the integrity and transparency of the charging process because all charging requests of all *ESUs* are recorded in a shared and immutable ledger, whose contents are verified by the blockchain validators through a predetermined consensus algorithm and cannot be altered. □

7. Performance Analysis

In this section, we evaluate our proposed scheme. First, we implement the scheme and evaluate it in terms of communication and computation overheads and this evaluation is called off-chain. Then, in the second part of the evaluation, on-chain, we demonstrate a proof-of-concept implementation of the blockchain side in our proposed scheme and assess its feasibility

7.1. Off-Chain

In this subsection, we evaluate the performance of our proposed scheme in terms of communication and computation overheads.

7.1.1. Communication Overhead

In this subsection, we calculate the size of all packets in our scheme to measure the communication overhead. We consider that the masked value of a charging request is 16 bytes, the timestamp is 8 bytes, and the size of an individual signature is 112 bytes. Table 2 summarizes the items sent from each ESU to the blockchain validators and their sizes. The calculated sizes of all packets are very small, which means that the communication overhead is reasonable.

Table 2. Communication overhead of our proposed scheme.

Data	Size "Bytes"
Charging Demand	16
Timestamp	8
Individual Signature	112

7.1.2. Computation Overhead

In this subsection, we evaluate the performance of our proposed scheme in terms of computational overhead. For the purpose of the experimental assessment, the simulation environment is set up to measure the computing time of the proposed scheme, in particular, the computing time for the composition of a charging request, the aggregation of the charging requests, and the verification of them. The details are as follows.

Environment setup: The simulation environment was set up in a Linux Ubuntu (64-bit) V20.04.3 LTS with an 11th gen intel(R) Core(TM) i5-1135G7@2.40 GHz processor and 16.0 GB memory. We used a supersingular elliptic curve with an asymmetric type 3 pairing size of 224 bits (MNT224 curve) for bilinear pairing and an $SHA-2$ hash function.

Simulation: In the simulation, the computation overhead was mainly due to the computation of several cryptosystems. To introduce an estimation for the next performance evaluation, we focused on the computation overhead of some of the cryptographic operations. Table 3 summarizes the time consumption of the fundamental cryptographic operations in our proposed scheme such as the scalar multiplication (M) in $\mathbb{G}_1$, pairing (P), hash function (H), exponentiation (E), and addition operation (A).

Table 3. The time consumption of the fundamental cryptographic operations.

Function	Time "ms"
Pairing	3.25
Hashing	0.04
Multiplication	0.00
Exponentiation	0.38
Addition	0.00

In the system initialization phase, all the operations that were run in this phase were offline and occurred every long period (week or month) and sometimes only once. This phase is operated by the *KMC* site and includes n exponentiation operations over $\mathbb{G}$ to compute R_i, calculating PV_i, which needs 2n exponentiation operations over $\mathbb{G}$ and n bilinear pairing operations $(3 \times n \times E) + (n \times P) + (2 \times n \times M)$. In the submission of charging requests phase, it is taken into consideration that all operations are carried out individually, and therefore, the overhead cost on the *ESUs* is $E + (2 \times H) + (3 \times M)$ to compute a CR message. In the third phase, upon receiving the charging requests, the aggregator aggregates the incoming charging request, verifies the aggregated signature, and computes the proof of the correct aggregation. The aggregation process needs a simple summation operation and its computation time can be ignored, whereas the verification of the aggregated signature takes $(3 \times P) + (2 \times n \times M) + (n+1) \times H$and the proof generation takes $(n \times M) + (n \times E)$, where n is the number of *ESUs*. In the final phase, the computing charging schedules phase, all operations in the phase can be ignored because they are offline. Table 4 summarizes the computation overhead of our proposed scheme in terms of phases, and Figure 5 summarizes the time consumed in terms of the *ESU* numbers for all phases.

Table 4. Computation overhead of our proposed scheme.

Phase	Entity	Total Operations	Computation Overhead "ms"
System Initialization Phase	*KMC* (Off-line)	$3nE + nP + 2nM$	$4.4n$
Submission Phase	*ESU*	$P + 2H + 3M$	0.47
Aggregation Phase	Aggregator	$3P + 3nM + nE + (n+1)H$	$0.08 + 0.42n$
Computing Charging Schedules Phase	*ESU*	Simple Comparison	-

In our proposed scheme we used VDAS, which performed well as the number of *ESUs* increased, as shown in Figure 5. In addition, we combined the VDAS scheme with a masking technique, which added a privacy layer by hiding the messages, with a negligible increase in the computation overhead.

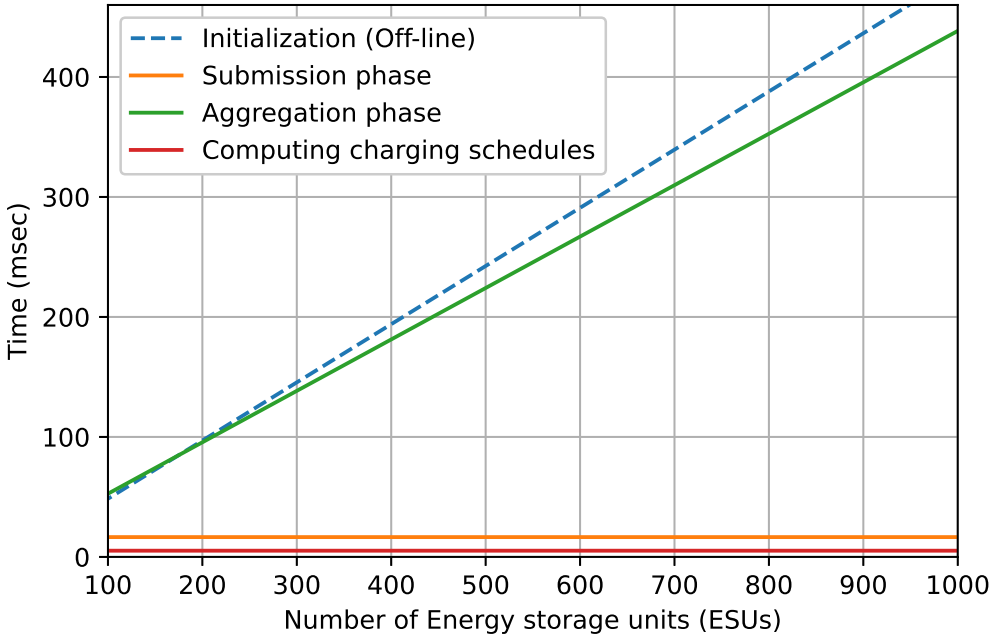


Figure 5. Time consumption vs. the number of *ESUs*.

Using the cryptographic operations and their computational times, we conveniently compared the computation time of several aggregate signature schemes' "individual signing phase" [15,42–46]. Figure 6 shows that the aggregate signature schemes CSZ [44] and VDAS [15] achieved better performance than the other aggregate signature schemes. However, the performance of CSZ depended on the number of *ESUs*, whereas CSZ performed well when the number of *ESUs* was less than or equal to five; otherwise, its computation overhead increased significantly with the number of *ESUs*.

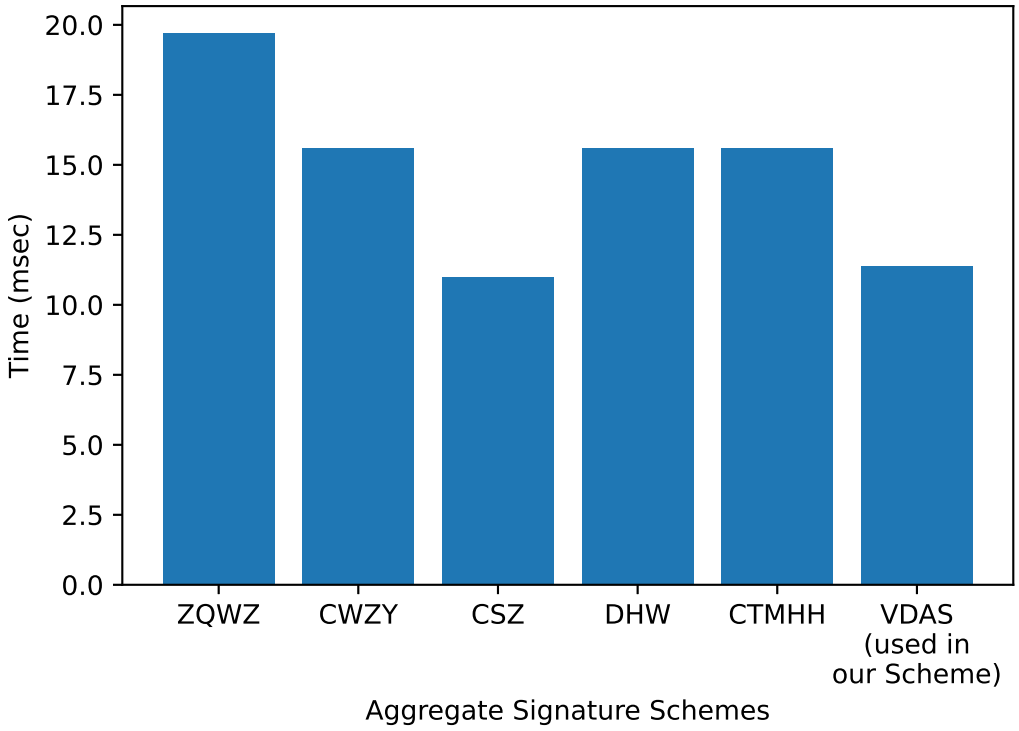


Figure 6. Comparing the aggregation signature schemes in terms of time.

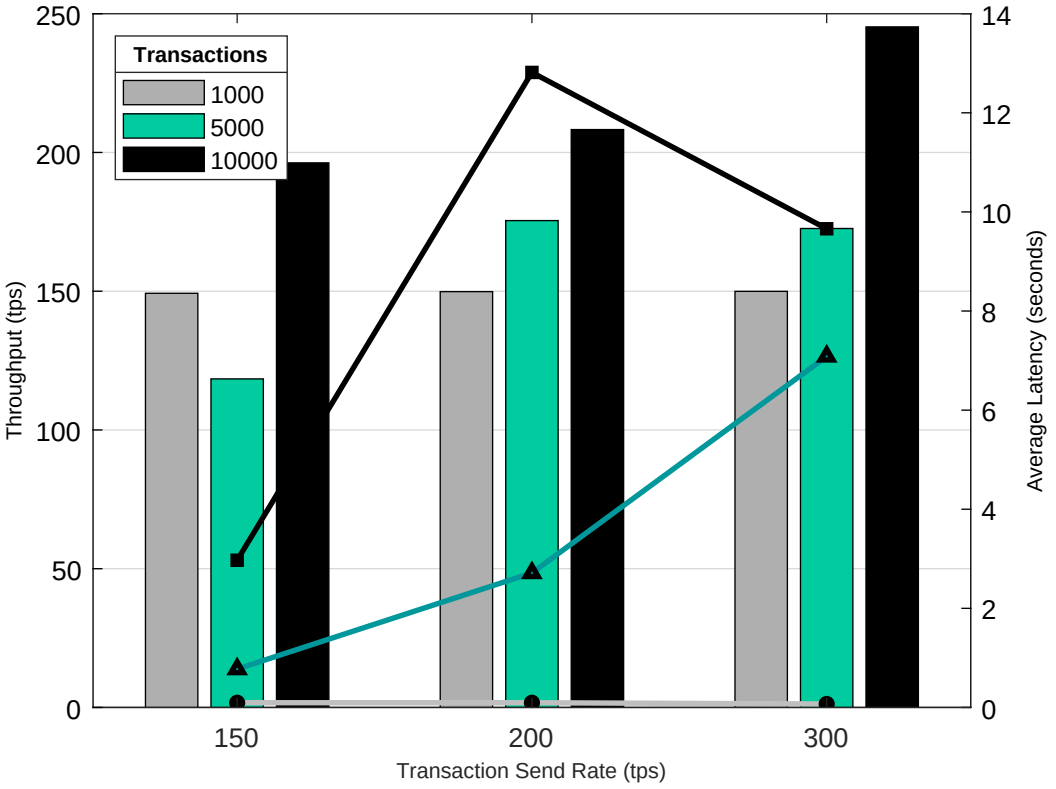
7.2. On-Chain Overheads

An overview of the proof-of-concept implementation of our proposed scheme was introduced using the hyperledger Caliper tool V0.3.2. For on-chain overheads, a chain code was implemented on the hyperledger fabric on an instance of Linux Ubuntu (64-bit) V20.04.3, and LTS was used on a DELL XPS 15 with a 2.20 GHz Intel Core i7 processor.

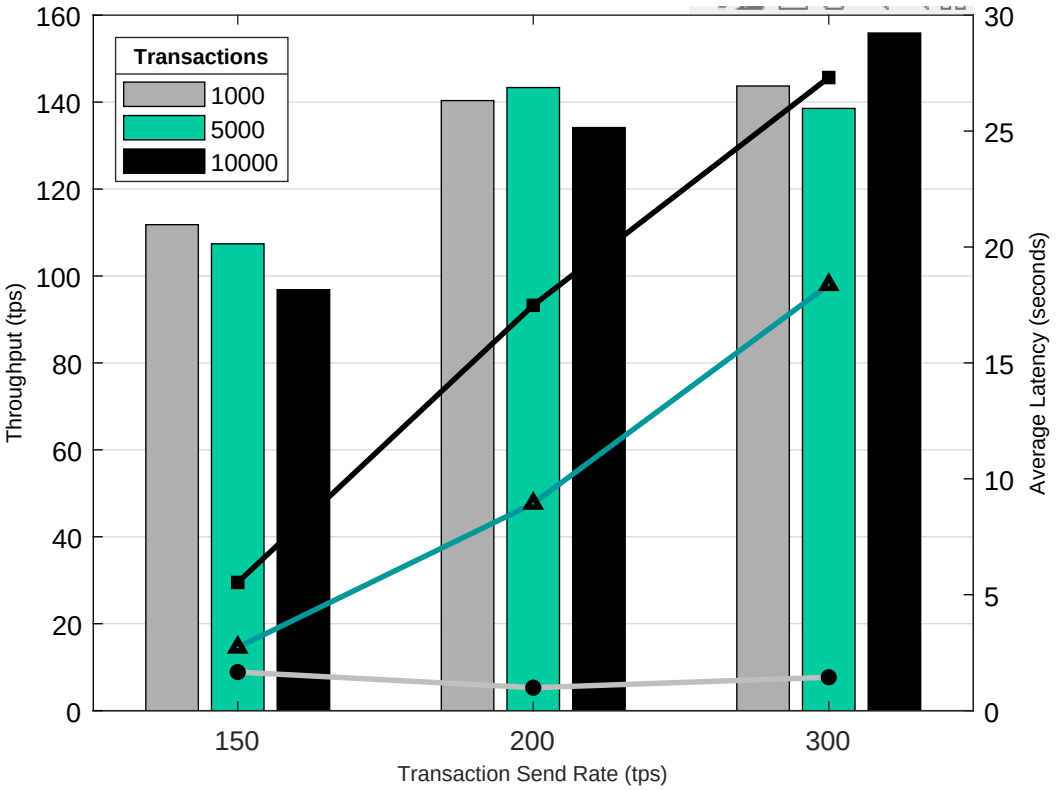
More specifically, we implemented a private blockchain that runs using different instances. Figure 7 shows the impact of varying the number of transactions (Δ) at different transactions per second (tps) on the throughput and latency. First, in the read transactions in Figure 7a, it can be seen that as tps increases, the throughput and latency remain fixed with Δ = 1000 (gray bar). However, with $\Delta = 5000$ and 10,000 (green and black bars), respectively, the throughput and latency increase. The green bar achieves its maximum throughput and latency with $tps = 200$ but the black bar achieves its maximum throughput and latency with $tps = 300$. Secondly, in the write transactions in Figure 7b, it can be seen that the three bars vary as tps increases, where with $tps = 150$, the maximum throughput and latency are performed by $\Delta = 1000$, whereas with $tps = 200$, the maximum throughput and latency are performed by $\Delta = 5000$. However, Δ = 10,000 achieves the maximum throughput and latency when $tps = 300$. This means that our scheme provides high throughput with an acceptable range of latency with high (Δ) in both reading and writing transactions.

Figure 8 shows the impact of varying the number of transactions at different transactions per second on the blockchain's error rates. Notice that when $\Delta = 1000$, the error rate remains constant at different tps in both read and write transactions. However, in the case of $\Delta = 5000$, the error rate remains fixed until $tps = 200$ in both read and write transactions. After that, the error rate increases slightly in read transactions and increases more in write transactions. When Δ = 10,000, the error rate remains fixed until $tps = 200$ and increases sharply in both read and write transactions.

It is found that if $\Delta = 1000$, any tps can work because in read transactions, the throughput, latency, and error rate remain constant. However, in write transactions, the throughput and latency increase as tps increases with a fixed error rate, whereas the throughput and latency perform better than other values with $tps = 150$ in write transactions. For $\Delta = 5000$, the maximum performance is achieved when $tps = 200$ in both read and write transactions, with minimum error rates in both cases. If Δ = 10,000, the maximum throughput and latency are achieved with $tps = 300$ but with maximum error rates in both read and write transactions. However, with $tps \leq 200$, the throughput and latency achieve maximum values relevant to the other Δ in read transactions only. We can see that our scheme is efficient in both reading and writing transactions even with very a high number of Δ with an acceptable range of latency and very small error rates.

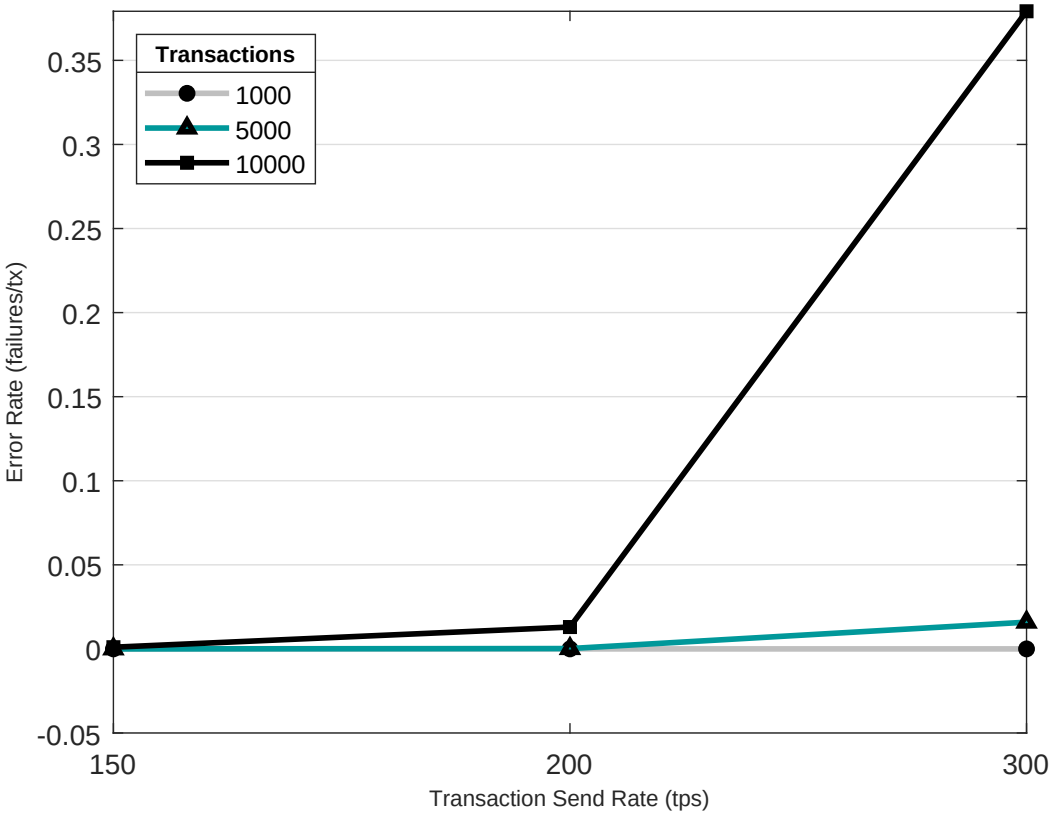


(**a**)

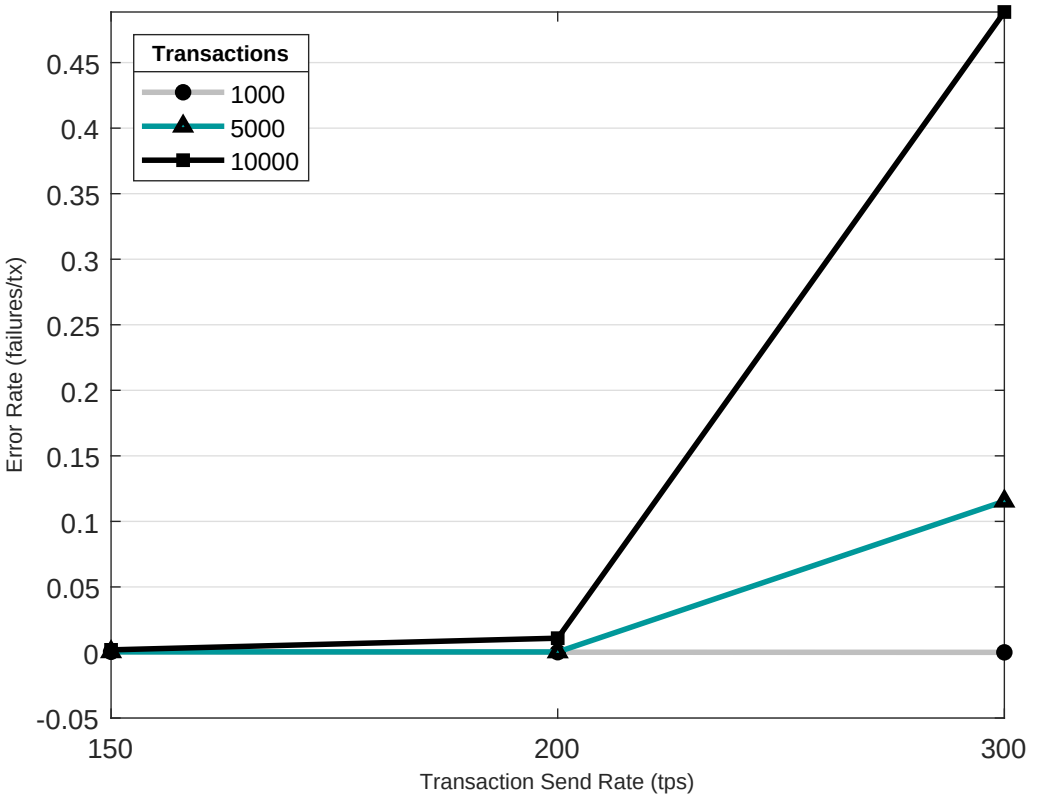


(**b**)

Figure 7. Results of the impact of varying the number of transactions at different transactions per second on the throughput and latency. (**a**) Read Transactions. (**b**) Write Transactions.



(**a**)



(**b**)

Figure 8. Results of the impact of varying the number of transactions at different transactions per second on the error rates. (**a**) Read Transactions. (**b**) Write Transactions.

8. Conclusions

In this paper, a privacy-preserving charging coordination scheme using a blockchain has been proposed. The blockchain is used to create a secure and transparent system to prevent a single point of failure and thwart *DoS* attacks. Privacy preservation is realized by a verifiable aggregation mechanism integrated with a masking technique and an aggregated signature technique to ensure the integrity of the received data and verify the identity of the *ESUs*. This verifiable aggregation mechanism enables the validator (local aggregator) to aggregate charging requests sent by the *ESUs* without knowing the individual charging requests to preserve privacy. Meanwhile, the correctness of the aggregation results can be checked by validators. In this way, malicious aggregators that send incorrect aggregation results can be identified. In addition, all messages sent by the *ESUs* are digitally signed to allow only the authorized *ESUs* to participate in the verifiable data aggregation process and the local aggregator to verify the integrity of the received data and authenticate the senders by verifying an aggregated signature instead of verifying the individual signatures to reduce the overhead. Security analysis, experiments, and simulations on both sides (off-chain and on-chain) were carried out to analyze the security of our scheme and evaluate its performance in terms of communication and computation overheads. The results confirm that the communication overhead in terms of message sizes is acceptable. We used a VDAS aggregate signature scheme, which was efficient, especially when increasing the number of *ESUs*. On the other hand, the on-chain experimental results indicated that our proposed scheme is efficient in both reading and writing transactions even with a very high number of transactions with acceptable latency and very small error rates. Finally, our proposed scheme preserves the privacy of consumers, the communication and computation overheads are acceptable, and the performance of the blockchain network is acceptable with a high number of *ESUs*.

Author Contributions: Conceptualization, H.H., M.M.E.A.M. and K.M.; methodology, H.H., M.M.E.A.M. and G.I.S.; software, H.H., M.B. and A.M.; validation, A.M. and G.I.S.; formal analysis, H.H. and K.M.; investigation, M.B., A.M. and G.I.S.; resources, M.M.E.A.M.; data curation, H.H.; writing—original draft preparation, H.H.; writing—review and editing, M.B. and M.M.E.A.M.; visualization, K.M.; supervision, G.I.S.; project administration, M.M.E.A.M.; funding acquisition, M.M.E.A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Qatar National Research Fund grant number NPRP13S-0201-200219.

Data Availability Statement: Not applicable.

Acknowledgments: This publication was made possible by NPRP grant # NPRP13S-0201-200219 from Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Habbak, H.; Metwally, K.; Mattar, A.M. Securing Big Data: A Survey on Security Solutions. In Proceedings of the 2022 13th International Conference on Electrical Engineering (ICEENG), Cairo, Egypt, 29–31 March 2022; pp. 145–149.
2. Wang, J.; Bharati, G.R.; Paudyal, S.; Ceylan, O.; Bhattarai, B.P.; Myers, K.S. Coordinated electric vehicle charging with reactive power support to distribution grids. *IEEE Trans. Ind. Inform.* **2019**, *15*, 54–63. [CrossRef]
3. Mahmoud, M.; Ismail, M.; Akula, P.; Akkaya, K.; Serpedin, E.; Qaraqe, K. Privacy-aware power charging coordination in future smart grid. In Proceedings of the 2016 IEEE Wireless Communications and Networking Conference, Dohar, Qatar, 3–6 April 2016; pp. 1–6.
4. Yang, Y.; Jia, Q.S.; Deconinck, G.; Guan, X.; Qiu, Z.; Hu, Z. Distributed Coordination of EV Charging with Renewable Energy in a Microgrid of Buildings. In Proceedings of the 2019 IEEE Power & Energy Society General Meeting (PESGM), Atlanta, GA, USA, 4–8 August 2019; p. 1.
5. Baza, M.; Nabil, M.; Ismail, M.; Mahmoud, M.; Serpedin, E.; Rahman, M.A. Blockchain-based charging coordination mechanism for smart grid energy storage units. In Proceedings of the 2019 IEEE International Conference on Blockchain (Blockchain), Atlanta, GA, USA, 14–17 July 2019; pp. 504–509.

6. Liu, L.; Zhou, K. Electric vehicle charging scheduling considering urgent demand under different charging modes. *Energy* **2022**, *249*, 123714. [CrossRef]
7. Yang, S.; Zhang, S.; Ye, J. A novel online scheduling algorithm and hierarchical protocol for large-scale EV charging coordination. *IEEE Access* **2019**, *7*, 101376–101387. [CrossRef]
8. Koufakis, A.M.; Rigas, E.S.; Bassiliades, N.; Ramchurn, S.D. Offline and online electric vehicle charging scheduling with V2V energy transfer. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 2128–2138. [CrossRef]
9. Baza, M.; Pazos-Revilla, M.; Sherif, A.; Nabil, M.; Aljohani, A.J.; Mahmoud, M.; Alasmary, W. Privacy-preserving and collusion-resistant charging coordination schemes for smart grids. *IEEE Trans. Dependable Secur. Comput.* **2021**, *19*, 2226–2243. [CrossRef]
10. Shafee, A.; Baza, M.; Talbert, D.A.; Fouda, M.M.; Nabil, M.; Mahmoud, M. Mimic learning to generate a shareable network intrusion detection model. In Proceedings of the 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 10–13 January 2020; pp. 1–6.
11. Yang, K.; Zhang, K.; Ren, J.; Shen, X. Security and privacy in mobile crowdsourcing networks: Challenges and opportunities. *IEEE Commun. Mag.* **2015**, *53*, 75–81. [CrossRef]
12. Li, M.; Weng, J.; Yang, A.; Lu, W.; Zhang, Y.; Hou, L.; Liu, J.N.; Xiang, Y.; Deng, R.H. CrowdBC: A blockchain-based decentralized framework for crowdsourcing. *IEEE Trans. Parallel Distrib. Syst.* **2018**, *30*, 1251–1266. [CrossRef]
13. Bhutta, M.N.M.; Khwaja, A.A.; Nadeem, A.; Ahmad, H.F.; Khan, M.K.; Hanif, M.A.; Song, H.; Alshamari, M.; Cao, Y. A survey on blockchain technology: Evolution, architecture and security. *IEEE Access* **2021**, *9*, 61048–61073. [CrossRef]
14. Li, T.; Gao, C.; Jiang, L.; Pedrycz, W.; Shen, J. Publicly verifiable privacy-preserving aggregation and its application in IoT. *J. Netw. Comput. Appl.* **2019**, *126*, 39–44. [CrossRef]
15. Liu, J.; Han, J.; Wu, L.; Sun, R.; Du, X. VDAS: Verifiable data aggregation scheme for Internet of Things. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; pp. 1–6.
16. Ota, Y.; Taniguchi, H.; Nakajima, T.; Liyanage, K.M.; Baba, J.; Yokoyama, A. Autonomous Distributed V2G (Vehicle-to-Grid) Satisfying Scheduled Charging. *IEEE Trans. Smart Grid* **2012**, *1*, 559–564. [CrossRef]
17. Das, S.; Acharjee, P.; Bhattacharya, A. Charging scheduling of electric vehicle incorporating grid-to-vehicle (G2V) and vehicle-to-grid (V2G) technology in smart-grid. In Proceedings of the 2020 IEEE International Conference on Power Electronics, Smart Grid and Renewable Energy (PESGRE2020), Kerala, India, 2–4 January 2020; pp. 1–6.
18. Huang, X.; Zhang, Y.; Li, D.; Han, L. An optimal scheduling algorithm for hybrid EV charging scenario using consortium blockchains. *Future Gener. Comput. Syst.* **2019**, *91*, 555–562. [CrossRef]
19. Tushar, W.; Zhang, J.A.; Smith, D.B.; Poor, H.V.; Thiébaux, S. Prioritizing consumers in smart grid: A game theoretic approach. *IEEE Trans. Smart Grid* **2014**, *5*, 1429–1438. [CrossRef]
20. Sortomme, E.; El-Sharkawi, M.A. Optimal scheduling of vehicle-to-grid energy and ancillary services. *IEEE Trans. Smart Grid* **2012**, *1*, 351–359. [CrossRef]
21. Kang, J.; Yu, R.; Huang, X.; Maharjan, S.; Zhang, Y.; Hossain, E. Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains. *IEEE Trans. Ind. Inform.* **2017**, *13*, 3154–3164. [CrossRef]
22. Tajeddini, M.A.; Kebriaei, H. A mean-field game method for decentralized charging coordination of a large population of plug-in electric vehicles. *IEEE Syst. J.* **2019**, *13*, 854–863. [CrossRef]
23. Tian, Z.; Jung, T.; Wang, Y.; Zhang, F.; Tu, L.; Xu, C.; Tian, C.; Li, X.Y. Real-time charging station recommendation system for electric-vehicle taxis. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 3098–3109. [CrossRef]
24. Cao, Y.; Tong, W.; Omprakash, K.; Geyong, M.; Naveed, A.; Abdullah, A. An EV Charging Management System Concerning Drivers' Trip Duration and Mobility Uncertainty. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *48*, 596–607. [CrossRef]
25. Xu, X.; Ke, D.; Li, L.; Xu, B. Optimal charging strategy for heterogeneous EVs for cyber-physical-social systems. In Proceedings of the 2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2), Beijing, China, 20–22 October 2018; pp. 1–5.
26. Akula, P.; Mahmoud, M.; Akkaya, K.; Songi, M. Privacy-preserving and secure communication scheme for power injection in smart grid. In Proceedings of the 2015 IEEE International Conference on Smart Grid Communications (SmartGridComm), Miami, FL, USA, 2–5 November 2015; pp. 37–42.
27. Alsharif, A.; Nabil, M.; Tonyali, S.; Mohammed, H.; Mahmoud, M.; Akkaya, K. EPIC: Efficient privacy-preserving scheme with EtoE data integrity and authenticity for AMI networks. *IEEE Internet Things J.* **2019**, *6*, 3309–3321. [CrossRef]
28. Tonyali, S.; Cakmak, O.; Akkaya, K.; Mahmoud, M.M.; Guvenc, I. Secure data obfuscation scheme to enable privacy-preserving state estimation in smart grid AMI networks. *IEEE Internet Things J.* **2015**, *3*, 709–719. [CrossRef]
29. Li, H.; Dan, G.; Nahrstedt, K. Lynx: Authenticated anonymous real-Time reporting of electric vehicle information. In Proceedings of the IEEE International Conference on Smart Grid Communications, SmartGridComm 2015, Miami, FL, USA, 2–5 November 2015; pp. 599–604.
30. Li, H.; Dan, G.; Nahrstedt, K. Portunes+: Privacy-Preserving Fast Authentication for Dynamic Electric Vehicle Charging. *IEEE Trans. Smart Grid* **2017**, *8*, 2305–2313. [CrossRef]
31. Fan, M.; Zhang, X. Consortium blockchain based data aggregation and regulation mechanism for smart grid. *IEEE Access* **2019**, *7*, 35929–35940. [CrossRef]
32. Li, H.; Han, D.; Tang, M. A privacy-preserving charging scheme for electric vehicles using blockchain and fog computing. *IEEE Syst. J.* **2020**, *15*, 3189–3200. [CrossRef]

33. Wang, Z. Identity-Based Verifiable Aggregator Oblivious Encryption and Its Applications in Smart Grids. *IEEE Trans. Sustain. Comput.* **2021**, *6*, 80–89. [CrossRef]
34. Baloglu, U.B.; Demir, Y. Lightweight privacy-preserving data aggregation scheme for smart grid metering infrastructure protection. *Int. J. Crit. Infrastruct. Prot.* **2018**, *22*, 16–24. [CrossRef]
35. Wang, X.D.; Meng, W.Z.; Liu, Y.N. Lightweight privacy-preserving data aggregation protocol against internal attacks in smart grid. *J. Inf. Secur. Appl.* **2020**, *55*, 102628. [CrossRef]
36. Badr, M.M.; Al Amiri, W.; Fouda, M.M.; Mahmoud, M.M.; Aljohani, A.J.; Alasmary, W. Smart parking system with privacy preservation and reputation management using blockchain. *IEEE Access* **2020**, *8*, 150823–150843. [CrossRef]
37. Shari, N.F.M.; Malip, A. State-of-the-art solutions of blockchain technology for data dissemination in smart cities: A comprehensive review. *Comput. Commun.* **2022**. [CrossRef]
38. Xie, J.; Tang, H.; Huang, T.; Yu, F.R.; Xie, R.; Liu, J.; Liu, Y. A survey of blockchain technology applied to smart cities: Research issues and challenges. *IEEE Commun. Surv. Tutorials* **2019**, *21*, 2794–2830. [CrossRef]
39. Alsunaidi, S.J.; Alhaidari, F.A. A survey of consensus algorithms for blockchain technology. In Proceedings of the 2019 International Conference on Computer and Information Sciences (ICCIS), Aljouf, Saudi Arabia, 3–4 April 2019; pp. 1–6.
40. Huo, R.; Zeng, S.; Wang, Z.; Shang, J.; Chen, W.; Huang, T.; Wang, S.; Yu, F.R.; Liu, Y. A Comprehensive Survey on Blockchain in Industrial Internet of Things: Motivations, Research Progresses, and Future Challenges. *IEEE Commun. Surv. Tutor.* **2022**. [CrossRef]
41. Zhao, N.; Zhang, G. Privacy-Protected Certificateless Aggregate Signature Scheme in VANET. In Proceedings of the 2019 11th International Conference on Wireless Communications and Signal Processing (WCSP), Xi'an, China, 23–25 October 2019; pp. 1–6.
42. Zhang, L.; Qin, B.; Wu, Q.; Zhang, F. Efficient many-to-one authentication with certificateless aggregate signatures. *Comput. Netw.* **2010**, *54*, 2482–2491. [CrossRef]
43. Chen, H.; Wei, S.M.; Zhu, C.; Yang, Y. Secure certificateless aggregate signature scheme. *J. Softw.* **2015**, *26*, 1173–1180.
44. Hu, C.; Wangan, S.; Bing, Z. Certificateless aggregate signature scheme. In Proceedings of the 2010 International Conference on E-Business and E-Government, Guangzhou, China, 7–9 May 2010; pp. 3790–3793.
45. Du Hong-zhen, H.M.j.; Wen, Q.y. Efficient and provably-secure certificateless aggregate signature scheme. *Acta Electonica Sin.* **2013**, *41*, 72.
46. Chen, Y.C.; Tso, R.; Mambo, M.; Huang, K.; Horng, G. Certificateless aggregate signature with efficient verification. *Secur. Commun. Netw.* **2015**, *8*, 2232–2243. [CrossRef]