frontiers | Frontiers in Communications and Networks

TYPE Original Research
PUBLISHED 30 April 2025
DOI 10.3389/frcmn.2025.1584152

OPEN ACCESS

EDITED BY
Patrick Hung,
Ontario Tech University, Canada

REVIEWED BY
Saadane Rachid,
École Hassania des Travaux Publics, Morocco
Nabil Sahli,
German University of Technology in Oman, Oman
Ibrahim Hajjeh,
Fransec, France

*CORRESPONDENCE
Mohamad Badra,
mohamad.badra@zu.ac.ae

RECEIVED 26 February 2025
ACCEPTED 21 April 2025
PUBLISHED 30 April 2025

CITATION
Badra M and Borghol R (2025) An efficient blockchain-based privacy preservation scheme for smart grids.
Front. Commun. Netw. 6:1584152.
doi: 10.3389/frcmn.2025.1584152

An efficient blockchain-based privacy preservation scheme for smart grids

Mohamad Badra[1]* and Rouba Borghol[2]

[1]Department of Computing and Applied Technology, College of Technological Innovation, Zayed University, Dubai, United Arab Emirates, [2]Department of Mathematics and Sciences, RIT Dubai, United Arab Emirates

Smart grids have revolutionized electricity management and distribution, but they also generate and transmit vast amounts of consumer data, raising privacy concerns. In this paper, we propose a blockchain-based solution to preserve user's privacy in smart grids and to mitigates data forgery, profiling, and man-in-the-middle attacks. Moreover, our solution provides security services such as authentication and non-repudiation to prevent unauthorized access to sensitive data and ensure accountability and traceability. We validate our approach through testing and show that it is a simple, scalable, cost-effective solution with minimal computational processing overhead.

KEYWORDS

blockchain, smart grid, privacy-preserving, smart contract, homomorphic encryption

1 Introduction

The ongoing revolutionary growth of communication and ubiquitous computing technologies has enabled the creation of innovative, intelligent applications and systems. One such system is the smart grid, designed to modernize the century-old power grid and to better meet the needs of the digital, eco-conscious society of the 21st century (Ma et al., 2013). The transformation from a traditional and provider-driven grid to a more consumer-engaged smart grid involves the implementation of bidirectional, Internet-based communication between the energy providers and their consumers.

The smart grid enables Utilities and consumers to exchange real-time information through bidirectional communications, facilitated by the Advanced Metering Infrastructure (AMI) (NIST Framework and Roadmap for smart grid interoperability standards, release 2.0, 2012). The AMI provides Utilities with new capabilities and functions to offer time-based rates, real-time energy measurement, and remote system control. To ensure accurate, up-to-date energy usage information, Utilities frequently aggregate data from smart meters installed at consumers' premises. A smart meter is a device that can perform useful and advanced functions such as time-based pricing, time-of-use billing, energy consumption reporting. Moreover, it can communicate with appliances installed at the consumer's premises to collect information of their electrical usage at regular intervals (NIST Framework and Roadmap for smart grid interoperability standards, release 2.0, 2012).

The two-way interaction between the Utility and the smart meters, combined with the vast amount of data generated by smart meters and collected by the Utility, can raise privacy concerns. In fact, the Utility can gain insights into the activities and behaviors within a consumer's home. This is enabled by the sensitive data that smart meters generate, related to the consumer's electricity usage patterns. Unauthorized parties may access and exploit this information, leading to significant privacy and security breaches. Failing to protect

customers' privacy could discourage them from adopting smart grids. Moreover, smart grids are vulnerable to various security risks inherited from IP-based communication infrastructures, such as spoofing, identity theft, man-in-the-middle attacks, and denial-of-service attacks (Hasan et al., 2022).

Several solutions have been proposed to preserve users' privacy in smart grids and to secure the data aggregated by the Utilities. In this paper, we present a hybrid approach, combining decentralized blockchain technology and additive homomorphic encryption to address the security and privacy issues in smart grids. Blockchain technology is an emerging field that is being actively explored in various systems and applications, including smart grids. It is a decentralized ledger that enables secure, transparent, and decentralized transactions, without the need for a central authority. By implementing a blockchain-based solution, we aim to enhance users' privacy and provide a secure and transparent way to store and manage users' data.

The organization of the paper is as follows. Section 2 describes privacy-preserving techniques in smart grids. Section 3 discusses recent approaches proposed for preserving the privacy of smart grid users. Section 4 describes the design and implementation of our proposed architecture. Section 5 evaluates the effectiveness of our solution in addressing privacy concerns in smart grids. Finally, our concluding remarks are presented in Section 6.

2 Privacy-preservation techniques in smart grids: balancing efficiency and security

Several techniques have been proposed to ensure privacy preservation in smart grids, particularly Data Aggregation (Wu et al., 2024), Homomorphic Encryption (Moore et al., 2014), Differential Privacy (Gai et al., 2020), Secure Multi-Party Computation (Yu et al., 2022), Anonymization, and Pseudonymization (Chen et al., 2023). Each of these techniques has its own advantages and disadvantages, particularly when balancing data privacy, system efficiency, and protection against data forgery and faults.

2.1 Data aggregation

Data aggregation is widely used as a privacy-preserving method in smart grids (Wu et al., 2024). Instead of transmitting individual users' data, aggregated data is collected and sent to grid operators. This helps reduce the risk of exposing individual consumption patterns. However, advanced statistical techniques can enable adversaries to infer individual consumption patterns, especially when combined with external datasets (e.g., consumption data, demographic data). Moreover, this technique can reduce the granularity of data, potentially hindering the ability to perform detailed analyses, provide tailored services to consumers, or to detect cyberattacks such as data forgery.

2.2 Homomorphic encryption

Homomorphic encryption allows specific types of computations to be carried out on ciphertext and obtains an encrypted result (Moore et al., 2014). For example, in the case of additive Homomorphic encryption, an entity who receives two encrypted messages would be able to decrypt the addition of two encrypted messages without being able to decrypt the messages individually. Although this technique provides strong encryption, ensuring privacy even during computation, it is computationally expensive and slower than working with unencrypted data. Moreover, it requires substantial resources for processing, limiting its scalability. Furthermore, using this technique without data authentication is ineffective at preventing man-in-the-middle attacks. An attacker can intercept and replace encrypted data sent from a particular user to the collector, with data encrypted using a random key selected by the attacker. Because this attacker-generated key is never shared with the key aggregator, the collector is unable to decrypt the combined encrypted requests it receives.

2.3 Differential privacy

Differential privacy (Gai et al., 2020) protects user privacy by adding random noise to data before it is shared or analyzed. This technique provides strong mathematical guarantees for safeguarding individual information. By introducing noise to datasets, such as energy consumption data, it becomes difficult for attackers to infer specific user patterns, even when analyzing aggregated information. However, adding too much noise can reduce data quality, making it less effective for smart grid applications. Moreover, implementing differential privacy in large-scale, real-time environments presents challenges due to its computational intensity and complexity.

2.4 Secure Multi-Party Computation (SMPC)

This technique enables multiple parties to perform computations on their private data without disclosing it to one another (Yu et al., 2022). In smart grids, it can be used to calculate energy consumption or billing information while keeping the underlying data private. Users share encrypted data and collaborate on computations, ensuring that no participant gains access to others' private information. However, as the number of participants increases, communication and computation demand grow exponentially, reducing the scalability of SMPC for large smart grids. Moreover, the need for extra cryptographic operations introduces significant computational overhead, making SMPC unsuitable for real-time applications in smart grids.

2.5 Anonymization and pseudonymization

Anonymization and pseudonymization protect privacy by altering or removing identifying information from energy consumption data (Chen et al., 2023). Anonymization removes all identifiers, making it impossible to trace data back to individuals, while pseudonymization replaces identifiers with pseudonyms that the data controller can re-identify under certain conditions. These techniques prevent direct identification of consumers by removing or substituting personal identifiers. However, anonymized data can sometimes be re-identified

through advanced methods, such as linking it to external datasets (Chen et al., 2023). Pseudonymization may also not work well in cases where personalized services or specific energy-saving recommendations are required in smart grids.

2.6 Blockchain-based solutions

Blockchain technology can enhance privacy in smart grids by storing energy consumption data on a decentralized, immutable ledger (Yin et al., 2023). When combined with cryptographic techniques like zero-knowledge proofs, blockchain ensures that only authorized users can access the data. Energy transactions, such as consumption and generation, are recorded on the blockchain, allowing users to verify their consumption or generation history without exposing private data. Zero-knowledge proofs enable users to demonstrate that they meet certain conditions (like energy consumption thresholds) without revealing specific details.

3 Related works

Overall, the development and implementation of effective privacy-preserving schemes play a crucial role in addressing privacy concerns and ensuring user confidence in smart grids. Various anonymous and privacy protection schemes have been proposed to prevent customer identification, with their effectiveness contingent on the specific context or architecture being utilized. Most of these schemes are primarily based on blind signatures, homomorphic encryption, trusted third parties, pseudonymity, differential privacy, and more.

In Khorasany et al. (2022), the authors present a blockchain-based mechanism for anonymous proof of location that verifies a participant's location while safeguarding the actual data. The level of privacy provided by this scheme depends on the number of public keys used to generate blockchain transactions. However, managing multiple keys introduces significant overhead and inefficiency. In Park et al. (2023), the authors propose a privacy-preserving scheme for aggregated data using blockchain, deep learning, and homomorphic encryption. The authors in Aitzhan and Svetinovic (2018) address privacy concerns in decentralized smart grids without relying on a trusted third party, but the solution's reliance on the Proof-of-Work consensus mechanism may hinder scalability and increase latency. In Maiti and Misra (2020), a solution suggests using proxy re-encryption to anonymously aggregate multi-dimensional data, while Tonyali et al. (2015) proposes a data obfuscation mechanism based on elliptic curve signature technology. However, these schemes fail to account for the computational costs associated with encryption, decryption, and signature computation, which increase with the number of users.

In Fotiou et al. (2021), a differential privacy model is proposed to create a marketplace for data, along with a blockchain-based solution for fair exchange and immutable data logs. However, the authors in Bracciale et al. (2022) identify security flaws in the scheme, particularly regarding user data confidentiality. Another differential privacy-based solution is presented in Hassan et al. (2020) to ensure that energy transaction queries do not reveal classified data or link transactions to real user identities.

In Li et al. (2021), an architecture based on a dual-blockchain system is discussed. One blockchain is private and is used to map associations between real and pseudonymous identities. The other is a shared blockchain, which ensures security and enables on-demand resource access.

The authors in Chen et al. (2022) introduce a double-blockchain solution to secure and anonymize the data aggregation process using Paillier encryption and batch signatures. In Bera et al. (2021), a private blockchain-based authorization scheme is proposed for secure and tamper-proof data transmission. This scheme demonstrates resilience against impersonation and replay attacks, as well as protection against man-in-the-middle and ephemeral secret leakage. However, as the number of peers in the blockchain network increases, the execution time of the scheme grows exponentially.

The authors in Mohammadali et al. (2018) discuss an identity-based key management protocol using Elliptic Curve Cryptography (ECC) to ensure user anonymity while maintaining low computational costs. However, this scheme is vulnerable to various attacks, as discussed in Mahmood et al. (2019), which proposes a pairing-based key management protocol as an alternative. Additionally, Kumar et al. (2019) presents a protocol focusing on data aggregation security and privacy but relies on a trusted third party to share a secret key between two users. Although the system incorporates correct Utility control initial verification and alternative solutions, such as those in Chaudhry et al. (2020), it lacks a provision for ensuring data integrity for the aggregated data.

In Liu et al. (2020), a scheme is presented to ensure privacy-preserving aggregation communication and function query for fog computing-based smart grids. The scheme minimizes system latency and communication overhead by leveraging edge computing resources. However, it centralizes user data storage at fog nodes or cloud servers, thereby inheriting the centralization issues typical of cloud-based schemes. In Zuo et al. (2020), the authors propose a scheme without a trusted authority in smart grids, based on the ElGamal homomorphic cryptosystem with distributed decryption. However, their scheme cannot guarantee the validity of data decryption when smart meters fail to work. In Zhang et al. (2021), a scheme enables an aggregator gateway to aggregate encrypted multi-type data and forward the aggregated data to the Utility. It provides integrity of aggregated data and preserves privacy. However, it lacks fault tolerance and error detection. In Chen et al. (2019), an elliptic curve-based scheme is presented. This scheme enables a meter to report multiple types of data at once. However, it fails to detect errors while decrypting aggregated data and suffers from some security threats such as data forgery and replay attacks.

3.1 Contributions of this work

To tackle the discussed challenges, we present a privacy-preserving scheme specifically tailored for the secure and reliable aggregation of data obtained from users' smart grids. By leveraging blockchain, smart contracts, and additive homomorphic encryption, our solution ensures the integrity and transparency of the aggregated

data and effectively mitigates various potential attacks, including replay attacks, man-in-the-middle attacks, data forgery, and secret key disclosure. Furthermore, our solution preserves user privacy, ensures non-repudiation, offers low-cost fault tolerance, enables authentication and confidentiality, and maintains long-term data integrity.

4 Our proposed solution

This section aims to describe our solution to ensure privacy in smart grid communications. The objective is to find an efficient solution that maintains smart grid performance and helps operators assess the necessity and risks of lacking privacy measures in communications. Our approach guarantees privacy, confidentiality, and authentication, while also offering transparency through the utilization of blockchain technology. Moreover, our solution provides user authentication and detects any anomalous data, while ensuring the immutability of each user transaction or request to the Utility.

Blockchain technology has been implemented to enhance transparency and provide protection against unauthorized modification of content, ensuring the existence of verifiable records (Bao et al., 2021). Smart contracts, acting as programmable agreements, enforce predefined rules to facilitate actions. Like real-world contracts, they are entirely digital and stored within distributed ledgers like blockchain. Smart contracts enable secure and transparent exchanges of money, property, or other assets, eliminating the need for intermediaries. They automate agreement execution, providing immediate certainty to all participants. Additionally, they can automate workflows by triggering subsequent actions based on predefined conditions. In the context of our paper, a smart contract includes a specific time window during which authorized users are allowed to insert their data into the smart contract.

To grant access to the Utility's system, the users need to register to the system during an initialization phase, and their public information is stored on both the Utility Data Center and the blockchain. Public information includes the user's key identifier of the secret key shared with the Utility, the user's public key encryption parameter, and the cluster's identifier to which the user belongs. The cluster forms a network of smart meters belonging to the same geographical region, such as those belonging to the same street or neighborhood. The Utility relies on the clusters for planning purposes. In fact, the Utility focuses on obtaining the total energy consumption of a cluster, instead of requiring information about individual smart meters' energy consumption.

In our proposed architecture, each user (designated as U_i) possesses a secret key shared with the Utility. Without the use of blockchain, users would be required to transmit their encrypted energy demands directly to the Utility or to an intermediary, which would then forward the aggregated requests to the Utility for decryption and processing. In this second scenario, the Utility and the intermediary could collaborate to discern the user's energy demand, thereby compromising the user's privacy. Moreover, relying on an intermediary exposes the system to security threats such as data forgery and replay attacks (Hasan et al., 2022).

In our proposed architecture, the Utility leverages blockchain technology to create smart contracts. The smart contracts are self-executing according to their algorithms, and they are accessed by authorized users to insert their data during a specific time window determined by the Utility. It is worth noting that the data inserted by users into a smart contract is accessible to all users on the blockchain. As part of our solution, we aim to address this issue by enabling the confidential insertion of data content into smart contracts. This data would be accessible in a clear and readable form only to its owner. Blockchain technology ensures data transparency, integrity, and availability. Hence, the Utility and any third party can access the data inserted by every user for later use (e.g., in case of dispute or to verify a transaction).

As illustrated in Figure 1 (step a), whenever the Utility updates its electricity rate, a new smart contract is generated and published to the blockchain, before sharing its reference with the users. Each interested user will then generate their energy demand D_i, a random value R_i known only to the user, and its secret key K_i shared with the Utility, contributing the sum to the smart contract as follows:

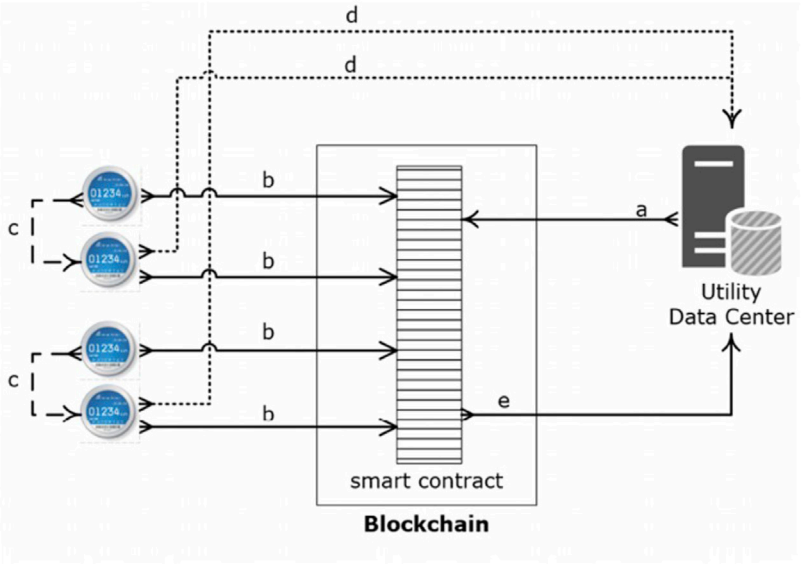
$$(D_i + U_i) + K_i \quad (1)$$

Every smart contract created by the Utility incorporates an integer value that is initially set to 0. When an authorized user successfully inserts their data into the smart contract, the integer value increases by one. The updated value is then used as an index to determine the user's position in the list of users who have inserted data. The inserted data includes, among other parameters, the user's identifier, and the user's public key (see Figure 1, step b).

Our solution implements ECDH (Elliptic Curve Diffie-Hellman) as a key agreement protocol between each pair of users. The algorithm's domain parameters are generated by the Utility and shared with each user during the initialization phase. During that same phase, each user generates a pair of public and private keys suitable for ECC. When a user inserts data into the smart contract, their public key will be appended to the inserted data.

The smart contract is essentially a program that acts as an agreement between parties. It combines the logic of the agreement with the code needed to enforce its terms automatically. The contract is stored on the blockchain, which makes it secure and tamper-proof. Each smart contract has a unique address, and its execution is triggered either by transactions sent to that address or by specific logical events. In fact, the smart contract can be programmed to execute after a specific time interval or at a predefined date and time. In our architecture, the smart meter displays energy usage, rewards, and grid demand data. Moreover, it handles aggregation, dynamic pricing, and settlement.

When the time window expires, each contributed user U_i who has been assigned an odd integer value (i.e., the index i is odd), will send its random value R_i to the next U_{i+1} in the list (see Figure 1, step c). The user U_i encrypts its random value R_i using a session key that is generated from the public keys of two users, following an ECDH key agreement protocol. Upon receipt, the user U_{i+1} deploys the same session key to decrypt the encrypted random value. This



Steps:
(a) Publishing the smart contract by the Utility; (b) Each user inserting their data into the smart contract (e.g., the smart meter); (c) Transmitting the random value from each user with an odd index to the next user with an even index; (d) Transmitting the sum of random values from users with an even index to the Utility; (e) The Utility reading the data inserted by users into the smart contract.

FIGURE 1
A simplified Architecture of our Proposed Solution.

process ensures the confidentiality and secure transmission of the random value between the two participating users.

Next, the user U_{i+1} symmetrically encrypts, using its secret key K_{i+1} shared with the Utility, the sum of random value R_i and of its own random value R_{i+1}. Then, it sends the result in (2) to the Utility (see Figure 1, step d), which symmetrically decrypts the result using the same key that is shared with U_{i+1} to obtain $(R_i + R_{i+1})$.

To generate the value in (2), U_{i+1} symmetrically encrypts, using its secret key K_{i+1} shared with the Utility, the sum of the random value R_i and its own random value R_{i+1}. This encrypted result is then sent to the Utility. Upon receipt, the Utility symmetrically decrypts, using the same shared key with U_{i+1}, the received data to obtain the sum $(R_i + R_{i+1})$.

$$R_i + R_{i+1} + K_{i+1} \tag{2}$$

In the case where the last user who inserted data into the smart contract has an odd integer index n, a specific procedure applies to the last three users in the list. The user U_{n-1} adds its own random value R_{n-1} to the random value R_{n-2} received from U_{n-2}. Next, U_{n-1} encrypts the result $(R_{n-1} + R_{n-2})$ using a session key shared with U_n and generated using an ECDH key agreement protocol. Upon receipt, U_n decrypts the encrypted data using the same session key to obtain $(R_{n-1} + R_{n-2})$. Next, U_n symmetrically encrypts the sum $(R_n + R_{n-1} + R_{n-2})$ using its secret key shared with the Utility. Then, the Utility decrypts it, using the same key to retrieve the original sum.

Any symmetric algorithm could be used, such as AES, to perform the encryption and decryption operations. However, for the sake of performance, our solution utilizes additive homomorphic encryption. This encryption method involves adding the keystream to the plaintext during encryption and subtracting the keystream during decryption. As discussed later, we incorporate protection against attacks such as data forgery and man-in-the-middle attacks by enabling HMAC (Hash-based Message Authentication Code).

After receiving and decrypting the sum of random values sent by every user with an even index, the Utility generates the sum of all random values, as follows:

$$\sum_{i=1}^{n} R_i \tag{3}$$

Next, the Utility uses the users' identifiers listed in the smart contract to retrieve their shared secret keys from its database (see Figure 1, step e) and then computes the sum of those keys, as follows:

$$\sum_{i=1}^{n} K_i \tag{4}$$

Afterwards, the Utility collects the sum in Equation 1 inserted by each user into the smart contract and computes their sum, as follows:

$$\sum_{i=1}^{n} (D_i + R_i + K_i) \tag{5}$$

Finally, the Utility adds the value in Equation 3 to the value in Equation 4 and subtracts the result from the value in Equation 5 to obtain the sum (Equation 6) of the users' energy demands, as follows:

$$\sum_{i=1}^{n}(D_i + R_i + K_i) - \left(\sum_{i=1}^{n}R_i + \sum_{i=1}^{n}K_i\right) = \sum_{i=1}^{n}D_i \quad (6)$$

5 Implementation and analysis

In this section, we evaluate and analyze our solution to demonstrate its effectiveness in preserving users' privacy and offering data integrity and confidentiality, non-repudiation, and low-cost fault tolerance. Furthermore, we assess its resilience against various attacks, including replay attacks, data forgery, man-in-the-middle attacks, and secret key disclosure. Moreover, we evaluate our solution in terms of computational cost and transmission overhead.

5.1 Privacy-preserving of users identities

In our solution, the privacy of the users is ensured by leveraging blockchain, cryptographic functions, and random values. At any moment, no one, including the Utility, can correlate the content of the user's energy demand with the user's identity. This is achieved by employing additive homomorphic encryption along with the inclusion of two random values before sharing the encrypted energy demands with the Utility.

The use of additive homomorphic encryption allows the Utility to perform computations on the encrypted data without decrypting it. As a result, the Utility cannot gain direct knowledge of the specific energy demand or link it to a particular user's identity. The inclusion of random values enhances privacy preservation by adding complexity, making it harder for the Utility to deduce any correlations between the request content and the user's identity.

At any moment, the Utility has the value in Equation 1 inserted by every user, as well as the sum of two random values received from a user with an even index. For example, the Utility can extract the value in Equation 1 of value U_1 and of value U_2 from the smart contract (i.e., $(D_1 + R_1 + K_1)$ and $(D_2 + R_2 + K_2)$. After getting $(R_1 + R_2)$ from $(U_2$, the Utility can apply (6) to obtain $(D_1 + D_2)$. However, there is no way for the Utility to determine D_1 or D_2 individually.

A potential privacy issue arises when user U_{i+1} and the Utility collaborate with each other. In this scenario, if U_i sends its encrypted random value R_i to U_{i+1}, the latter may act maliciously and share R_i with the Utility. This action could potentially disclose not only the energy demand D_i of U_i but also the energy demand D_{i+1} of U_{i+1}. The sharing of R_i with the Utility by U_{i+1} compromises the privacy of both U_i and U_{i+1}. This is because U_i's encrypted random value R_i is used in the computation of U_{i+1}'s energy demand D_{i+1}. By revealing R_i, the Utility can potentially link R_i to U_i's identity and infer information about D_i and D_{i+1}. However, we assume that this type of collaboration is unlikely to occur.

5.2 Data transmission integrity and data forgery detection

Our solution provides comprehensive protection against not only data breaches but also unauthorized modifications or tampering of data transmitted from users to the Utility. We understand that man-in-the-middle attacks pose a potential risk, where attackers can intercept or forge the encrypted data transmitted from every user with an even index to the Utility. Our solution addresses this concern in a flexible and straightforward manner. It incorporates HMAC applied to a timestamp, the sum of two random values, and the secret key shared between the Utility and the user with an even index. The user then sends both its value in Equation 2 and the HMAC output to the Utility.

Upon receipt, the Utility can verify the integrity of the value in Equation 2 ensuring that the data has not been altered or modified during its transmission. This provides robust protection against potential man-in-the-middle attacks, error detection and data forgery. It is worth noting that this HMAC-based security measure incurs negligible performance overhead compared to all existing measures.

5.3 Non-repudiation of energy demand and long-term data integrity

Non-repudiation of demand refers to the assurance that a user cannot deny generating a specific energy demand. It ensures that once a user has submitted their energy demand to the Utility or smart grid system, they cannot later deny having made that request.

Our solution ensures the non-repudiation of demand. With the sum of the two random values and their secret keys, the Utility can calculate the total energy demands inserted by both users into the smart contract. In case of dispute with either user, the Utility can request that U_i or U_{i+1} disclose the value of its respective energy demand. This ensures that neither user can deny the amount of their energy demand, as it can be verified through their disclosed random values and the computed sum. By leveraging blockchain technology, our solution can maintain long-term data integrity by storing information in an immutable and tamper-proof manner. Hence, a user who has inserted an energy demand into the smart contract cannot deny it at a later stage.

It is worth noting that several previous works have addressed the issue of multiparty non-repudiation by employing various techniques. These include (Hasan et al., 2022) PKI-based solutions (Public Key Infrastructures), electronic notary systems, trusted third-party (TTP) mechanisms, ID-based non-repudiation approaches, and more (Li et al., 2022). However, they require significant computational operations, considerable transmission overhead, and complex management processes.

5.4 Data confidentiality and transparency

Before including its energy demand into a smart contract, the user encrypts, using a secret key shared with the Utility, the sum of its energy demand and a random value. This encryption ensures that the energy demand remains confidential and protected from unauthorized access. The encryption process takes three inputs to generate the value in Equation 1: the user's energy demand, a random value only known to the user, and the secret key shared with the Utility. Without knowing the random value, it is very difficult for anyone to decrypt the value in Equation 1. This enhances

the security and privacy of the user's data during the data insertion process.

To transmit its random value to the next user in the list, the two users employ an ECDH key agreement protocol to generate a session key. That session key will be used to encrypt the random value before transmitting it to the user with an even index.

To transmit the sum of two random values to the Utility, the user employs additive homomorphic encryption. This encryption takes two inputs to generate the encrypted output: the sum of the random values and the secret key shared between the Utility and the user with an even index. Additive homomorphic encryption ensures the confidentiality of the sum of random values during transmission. Moreover, it allows the Utility to perform calculations on the encrypted data without compromising the privacy of the individual random values.

In all transmitted messages, users include their identifiers shared with the Utility. However, for the sake of simplicity, we omitted this detail from the description.

As for data transparency, the blockchain is inherently transparent, allowing anyone to view both the transactions and the code of the smart contracts. Moreover, anyone can review the code to confirm that the contract will execute as intended. On the other hand, transactions are permanently recorded on the blockchain, providing a permanent record of actions and agreements. The public nature of smart contract execution promotes accountability and helps resolve disputes or verify transactions.

5.5 Detection of replay attacks

Our solution is designed to be resilient to replay attacks, which occur when an attacker intercepts and maliciously retransmits data packets. We prevent such attacks by including an HMAC applied to the data, along with a timestamp. The inclusion of the timestamp allows us to easily detect and discard outdated or duplicate data packets, effectively preventing replay attacks.

5.6 Cheap fault tolerance

We define fault tolerance as the ability of the Utility to continue functioning properly and provide reliable services even in the presence of faults or failures from specific users. Even if a few users fail to submit their energy values in Equation 2, the Utility can still obtain the remaining energy demands from other operational users. This allows the Utility to continue computing the sum of the received random values, thereby maintaining reliable services.

If a user fails to send its encrypted value to the user with an even index, the Utility will exclude their value. Similarly, if the latter user fails to send the encrypted sum of the random values to the Utility, the Utility will exclude both users' values. These values will be excluded from the smart contract when computing the value in Equation 5.

If a user fails to send its encrypted value to the user with an even index, the Utility will exclude their value. Similarly, if the latter user fails to send the encrypted sum of the random values to the Utility, their value will also be excluded. These values will be omitted from the smart contract when computing the value in Equation 5.

This ensures that the Utility can obtain a comprehensive overview of the energy demands from operational users while disregarding values from failed ones. As a result, the system remains functional and resilient, even in the presence of user failures. It is worth noting that the Utility can track failed users and take appropriate actions, such as removing them from the cluster if they repeatedly fail to complete their tasks.

5.7 Secret key disclosure

Preventing secret key disclosure is crucial for maintaining the security of cryptographic systems. This involves implementing robust key management practices, including secure storage, controlled access, and secure transmission of keys.

According to the design of our solution, the disclosure of the secret keys shared between the users and the Utility does not compromise the overall security of our architecture. As we mentioned earlier, when a user sends its random value to the user with an even index, both users must first generate a session key based on their ECDH public keys. This session key will be used as an input for the encryption/decryption process.

Although the secret key shared between a user and the Utility could be disclosed, an adversary would still need the victim's private key. This is necessary to generate the session key and impersonate the victim's identity or perform actions on their behalf. In this regard, our solution is resilient to secret key disclosure.

5.8 Collaborative risks and resilient aggregation

Our solution stands in sharp contrast to methods that require key updates or key redistribution among users. It offers a flexible user enrollment and revocation mechanism, allowing users to join or leave the data aggregation process dynamically.

In scenarios involving collaborative risks, several issues should be considered, especially potential Sybil attacks and communication latency in large-scale networks. Malicious entities could create multiple fake identities to manipulate the system or disrupt its functionality. Our solution mitigates Sybil attacks directly through the usage of HMAC applied to each user request. While communication overhead is optimized by our solution, delays may still occur in wide-area deployments, especially in areas with limited network infrastructure. These delays could affect the timeliness of energy demand aggregation. Grouping users into clusters improves performance but may face limitations in dynamic environments where users frequently join or leave clusters. Since our solution leverages blockchain, it does not require re-clustering mechanisms to manage resource allocation in clusters, thus avoiding significant delays or overheads.

While some existing solutions offer strong privacy guarantees, they often assume users within a cluster act honestly when sharing their encrypted energy demands. If a user deliberately shares incorrect values, it could compromise the integrity of aggregated demands. In contrast, the impact of such behavior in our solution is

limited: only the user providing the value in (2) and the preceding user in the list will be affected if one of them shares incorrect values. The demands of other users remain unaffected, ensuring greater resilience.

5.9 Proof of concept

As a proof of concept, we have developed two straightforward Python programs in conjunction with Solana for the blockchain network. These programs demonstrate the achievement of privacy preservation for users. One program represents the Utility, and the other represents the consumers (users). The Utility program serves as the coordinator, responsible for announcing electricity prices and creating corresponding smart contracts on the Solana blockchain. When a new price is determined, the program deploys a smart contract that encapsulates the auction logic and participation rules. The contract reference (a unique identifier) is then shared with potential participants.

The users can optionally participate in the process. Each user registers for the auction by inserting its value (as per Equation 1) into the smart contract. This ensures that user data remains private while leveraging the blockchain's transparency and immutability.

This proof of concept demonstrates the effective use of blockchain technology, specifically Solana, for decentralized and privacy-preserving auctions. It highlights the role of smart contracts as secure intermediaries for data submission and auction management while maintaining user confidentiality.

Blockchain was selected over Federated Learning due to its enhanced security, decentralization, and transparency. In fact, blockchain better aligns with the requirement for decentralized decision-making and maintaining user privacy, without the need to a collaborative model training. Moreover, blockchain natively supports transaction-based processes and smart contract functionality, which are central to the described system. Blockchain eliminates the need for a trusted central server, ensuring transparency and reducing the risk of data manipulation. In addition, blockchain ensures scalability and efficiency for decentralized auctions.

Our proposed solution is designed to balance privacy, security, and transparency in smart grid communications. While scalable to an extent, challenges related to computational overhead, transaction throughput, and consensus mechanisms need to be addressed to support large-scale deployments effectively. Implementing targeted optimizations and leveraging advanced blockchain technologies can significantly enhance the system's scalability, ensuring reliable performance even in expansive smart grid networks.

It is worth noting that real-time data flow and quick responses to changes, like sudden demand spikes, faults, or shifts in energy distribution, are vital for keeping the grid stable and efficient. In addition, the advanced data protection can slow down real-time decision-making, while not enough can leave the system open to security threats. To address this, encryption methods need to be more efficient, or faster processing techniques should be adopted, so the grid can respond quickly while keeping data secure and intact. On the other hand, the key management system must be carefully designed to prevent key exposure or misuse. Finally, the time window for data insertion and the encryption/decryption cycles could delay the ability to quickly adjust to dynamic grid conditions, which is critical in smart grid management.

The scalability of our proposed solution relies heavily on efficient data handling, privacy-preserving encryption techniques, and robust blockchain architecture. As the system scales, performance may be affected by increased transaction volumes and computational loads. However, our solution maintains the system's performance, security, and privacy as the number of users grows. In fact, it is based on optimizing blockchain architecture, leveraging distributed computing, and implementing lightweight encryption techniques.

From storage perspective, our proposed solution requires storing user data, public keys, and transaction details in the form of smart contracts. Public information, such as public keys and identifiers, is stored for each user. The overhead depends on the size of cryptographic keys and metadata for each transaction. As the number of users and transactions grows, blockchain storage can quickly become a bottleneck, especially if there are no efficient pruning mechanisms in place.

According to Figure 2, the system shows improved efficiency in handling users compared to the initial 1:1 user-to-core ratio. In fact, only 3.5 cores are needed to support 1000 users. This suggests that the system scales effectively. The enhanced scalability is likely due to optimizations such as load balancing, multithreading, and resource sharing. These optimizations improve core utilization and performance as the user load increases.

Although this configuration can be optimal for real-life scenarios, further optimization can be achieved by following the recommendations below:

- Rewrite the application using Rust: Rust is a high-performance language known for its efficiency, memory safety features, and excellent memory management capabilities.
- Profile memory usage: Use Rust's profiling tools to analyze and optimize memory allocation within the application. This will help identify any memory inefficiencies, enabling better memory management.
- Ensure proper data encoding and decoding: Leverage Rust's strong type system to ensure proper data encoding and decoding. This minimizes unnecessary data transformations and improves overall performance.
- Utilize Rust's standard library: Leverage the efficient networking tools and data structures provided by Rust's standard library. This will enhance the application's ability to efficiently manage and handle high workloads.

5.10 Analysis of computational cost and communication overhead

For performance optimization and to eliminate delays associated with session key generation, the Utility can group users into clusters, each containing a few hundred users. During the transmission of random values, both users should agree on a session key using ECDH. To avoid repeated session key generation for future transmissions, each user stores the session key along with the other user's identifier in a personal data structure.

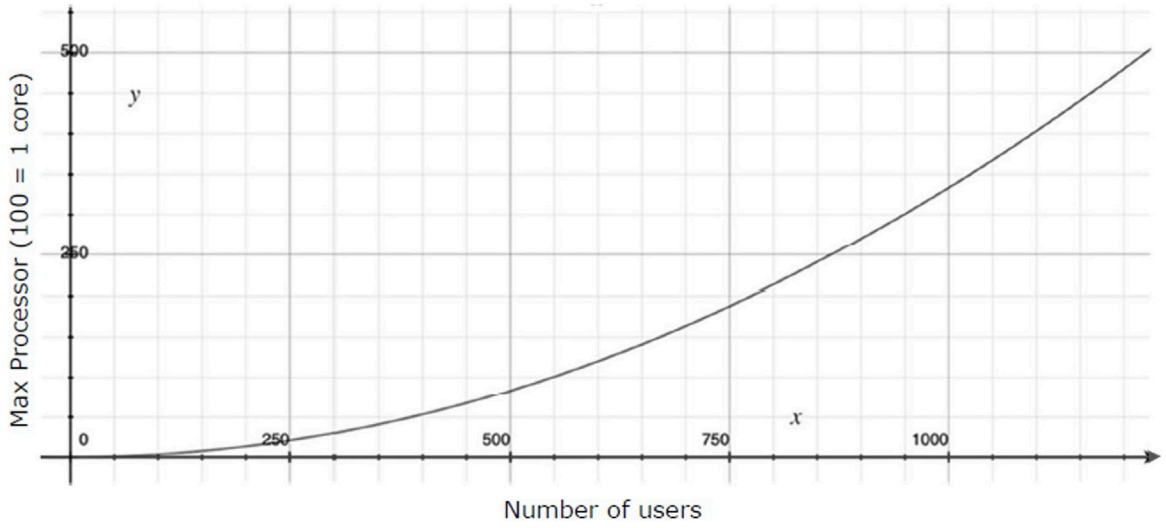


FIGURE 2
Number of core processors to handle users' requests.

TABLE 1 Computational cost.

Entity	Homomorphic encryption	AES
Utility	$\frac{n*(T_{H_{dec}}+T_{HMAC})}{2}$	$\frac{n*(T_{H_{enc}}+T_{H_{dec}}+T_{HMAC})}{2}$
User	$\frac{T_{H_{enc}}+T_{H_{dec}}+T_{HMAC}}{2}$	$\frac{T_{enc}+T_{dec}+T_{H_{enc}}}{2}$

n represents the number of users.
$T_{H_{enc}}$ represents the computation for homomorphic encryption.
$T_{H_{dec}}$ represents the computation for homomorphic decryption.
T_{enc} represents the computation for symmetric encryption.
T_{dec} represents the computation for symmetric decryption.
T_{HMAC} represents the computation for HMAC.

Alternatively, during the initialization phase, the Utility can share the ECDH public keys of all users within a cluster. Each user can then generate session keys offline for use with other users within the same cluster. This approach eliminates the need to include the user's ECDH public key in their energy demands and avoids the requirement of generating session keys dynamically.

In practice, the Utility can create clusters with up to 100 users. In this case, each user would only need 1.6 KB of storage to store the session keys. These session keys, along with ECDH private keys and individual secret keys shared with the Utility, can be securely stored in tamper-resistant devices, such as smart cards or TPMs (Trusted Platform Modules).

Table 1 shows the computational costs of our solution in a cluster consisting of n users. As previously mentioned, users can choose between AES or homomorphic encryption when transmitting their random values to the user with an even index. In the AES encryption scenario, calculating HMAC is not required. However, in the homomorphic encryption scenario, HMAC is necessary to mitigate the risks of data forgery.

In terms of communication overhead, a total of $(n + 1)$ messages are needed to add the smart contract to the blockchain and insert the users' energy demands. Half of these messages are transmitted locally over a Neighborhood Area Network (NAN), resulting in minimal delays. The other half are sent to the Utility over a Wide Area Network (WAN). Moreover, $(n + 1)$ messages are required to convey the random values between users with even indices and the Utility. However, the second set of $(n + 1)$ messages should not be counted as additional overhead, as they are part of the normal energy demand aggregation process. Therefore, only $(n + 1)$ messages should be considered as communication overhead in the implementation of our solution. The communication overhead and computational costs of our solution are negligible compared to existing solutions. Furthermore, our solution ensures all the security services provided by those solutions, along with the added security properties previously discussed.

When comparing our solution with existing ones, using metrics like communication and computation overhead, the results show that our scheme outperforms or matches the efficiency of others. Moreover, our solution addresses several cybersecurity issues that others cannot.

Table 2 presents the comparison results across schemes (Liu et al., 2020; Zuo et al., 2020; Zhang et al., 2021; Chen et al., 2019), covering the communication costs and the following key security features: anonymity, fault tolerance, replay attack prevention, error detection, non-repudiation, tamper resilience, and data forgery resilience. The results demonstrate that our scheme offers superior security features compared to existing schemes.

6 Conclusion

This paper proposes a new blockchain-based, privacy-preserving solution for users in the smart grid. It is designed to mitigate various attacks, including data forgery, man-in-the-middle, and replay attacks. Moreover, it efficiently reduces computational costs and communication overhead, making it well-suited for resource-constrained devices like smart meters.

We provide comprehensive analyses concerning security and performance comparisons and demonstrate significant advantages

TABLE 2 Security features and communication costs comparison.

Schemes	Anonymity	Fault-tolerance	Anti-replay	Error-detection	Non-repudiation	Tamper-resilience	Anti-forgery	Communication costs
Liu et al. (2020)	✓	✗	✗	✗	✗	✓	✓	High
Zuo et al. (2020)	✗	✗	✓	✗	✗	✓	✓	Medium
Zhang et al. (2021)	✗	✗	✓	✗	✗	✓	✓	High
Chen et al. (2019)	✗	✗	✓	✗	✗	✓	✓	Medium
Ours	✓	✓	✓	✓	✓	✓	✓	Medium

over existing schemes. The solution also supports additional security services, including non-repudiation and long-term data integrity. As a result, it holds great promise for ensuring the privacy of smart grid users.

To further enhance the usability of our scheme, we plan to focus on the performance and efficiency of blockchain and assess its impact on our solution's performance.

Data availability statement

The original contributions presented in the study are included in the article. Further inquiries can be directed to the corresponding author.

Author contributions

MB: Conceptualization, Data curation, Formal Analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review and editing. RB: Conceptualization, Data curation, Formal Analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review and editing.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. Financial support was received from Zayed University and RIT Dubai.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Generative AI was used in the creation of this manuscript.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Aitzhan, N. Z., and Svetinovic, D. (2018). Security and privacy in decentralized energy trading through multi-signatures blockchain and anonymous messaging streams. *IEEE Trans. Dependable Secure Comput.* 15, 840–852. doi:10.1109/tdsc.2016.2616861

Bao, J., He, D., Luo, M., and Choo, K. K. R. (2021). A survey of blockchain applications in the energy sector. *IEEE Syst.* 15, 3370–3381. doi:10.1109/jsyst.2020.2998791

Bera, B., Saha, S., Das, A. K., and Vasilakos, A. V. (2021). Designing blockchain-based access control protocol in IoT-enabled smart-grid system. *IEEE Internet Things* 7, 5744–5761. doi:10.1109/jiot.2020.3030308

Bracciale, L., Raso, E., Gallo, P., Sanseverino, E. R., Bianchi, G., and Loreti, P. (2022). "Privacy in blockchain-based smart grids," in *Proceedings of the workshop on blockchain renewables integration (BLORIN)* (Palermo, Italy), 37–41.

Chaudhry, S. A., Yahya, K., and Al-Turjman, F. (2020). *Correctness of an authentication scheme for managing demand response in smart grid.* Smart-Grid in IoT-enabled spaces. New York, NY, USA: Taylor & Francis, 223–231.

Chen, M., Shan, C., Chang, Z., Iqbal, M., and Almakhles, D. (2023). Data anonymization evaluation against re-identification attacks in edge storage. *Wirel. Netw.* 30, 5263–5277. doi:10.1007/s11276-023-03235-6

Chen, S., Yang, L., Zhao, C., Varadarajan, V., and Wang, K. (2022). Double-blockchain assisted secure and anonymous data aggregation for fog-enabled smart grid. *Engineering* 8, 159–169. doi:10.1016/j.eng.2020.06.018

Chen, Y., Martinez-Ortega, J. F., Castillejo, P., and Lopez, L. (2019). An elliptic curve-based scalable data aggregation scheme for smart grid. *IEEE Syst. J.* 14, 2066–2077. doi:10.1109/jsyst.2019.2954080

Fotiou, N., Pittaras, I., Siris, V. A., Polyzos, G. C., and Anton, P. (2021). A privacy-preserving statistics marketplace using local differential privacy and blockchain: an application to smart-grid measurements sharing. *Blockchain Res. Ap-plications* 2, 100022. doi:10.1016/j.bcra.2021.100022

Gai, N., Xue, K., Zhu, B., Yang, J., Liu, J., He, D., et al. (2020). "An efficient data aggregation scheme with local differential privacy in smart grid," in *Proceedings of 16th*

international conference on mobility, sensing and networking (MSN) (Tokyo, Japan), 73–80. doi:10.1109/MSN50589.2020.00027

Hasan, M. K., Habib, A. A., Shukur, Z., Ibrahim, F., Islam, S., and Razzaque, M. A. (2022). Review on cyber-physical and cyber-security system in smart grid: standards, protocols, constraints, and recommendations. *Netw. Comput. Appl.*, 209. doi:10.1016/j.jnca.2022.103540

Hassan, M. U., Rehmani, M. H., and Chen, J. (2020). DEAL: differentially private auction for blockchain-based microgrids energy trading. *EEE Trans. Serv. Comput.* 13, 263–275.

Khorasany, M., Dorri, A., Razzaghi, R., and Jurdak, R. (2022). Lightweight blockchain Framework for location-aware peer-to-peer energy trading. *Int. J. Electr. Power and Energy Syst.*, 127. doi:10.1016/j.ijepes.2020.106610

Kumar, N., Aujla, G. S., Das, A. K., and Conti, M. (2019). ECCAuth: a secure authentication protocol for demand response management in a smart grid system. *IEEE Trans. Industrial Inf.* 15, 6572–6582. doi:10.1109/tii.2019.2922697

Li, F., Li, X., Liu, P., Sun, X., Yu, S., and Ge, J. (2022). A privacy-aware electricity consumption data collection model based on group blind signature. *Secur. Commun. Netw.* 2022, 1–14. doi:10.1155/2022/4352291

Li, K., Yang, Y., Wang, S., Shi, R., and Li, J. (2021). A lightweight privacy-preserving and sharing scheme with dual-blockchain for intelligent pricing system of smart grid. *Comput. Secur.*, 103. doi:10.1016/j.cose.2021.102189

Liu, J. N., Weng, J., Yang, A., Chen, Y., and Lin, X. (2020). Enabling efficient and privacy-preserving aggregation communication and function query for fog computing based smart grid. *IEEE Trans. Smart Grid* 11, 247–257. doi:10.1109/tsg.2019.2920836

Ma, R., Chen, H. H., Huang, Y. R., and Meng, W. (2013). Smart grid communication: its challenges and opportunities. *IEEE Trans. Smart Grid* 4, 36–46. doi:10.1109/tsg.2012.2225851

Mahmood, K., Arshad, J., Chaudhry, S. A., and Kumari, S. (2019). An enhanced anonymous identity-based key agreement protocol for smart grid advanced metering infrastructure. *Int. J. Commun. Syst.* 32. doi:10.1002/dac.4137

Maiti, S., and Misra, S. (2020). P2B: privacy preserving identity-based broadcast proxy Re-encryption. *IEEE Trans Veh. Technol.* 69, 5610–5617. doi:10.1109/tvt.2020.2982422

Mohammadali, A., Haghighi, M. S., Tadayon, M. H., and Mohammadi-Nodooshan, A. (2018). A novel identity-based key establishment method for advanced metering infrastructure in smart grid. *IEEE Trans. Smart Grid* 9, 2834–2842. doi:10.1109/TSG.2016.2620939

Moore, C., O'Neill, M., O'Sullivan, E., Doröz, Y., and Sunar, B. (2014). "Practical homomorphic encryption: a survey," in *Proceedings of 2014 IEEE international symposium on circuits and systems (ISCAS)* (Melbourne, VIC, Australia), 2792–2795. doi:10.1109/ISCAS.2014.6865753

NIST Framework and Roadmap for smart grid interoperability standards, release 2.0 (2012). NIST Special Publication 1108R2.

Park, K., Lee, J., Das, A., and Park, Y.BPPS (2023). BPPS:Blockchain-Enabled privacy-preserving scheme for demand-response management in smart grid environments. *IEEE Trans. Dependable Secure Comput.* 20, 1719–1729. doi:10.1109/tdsc.2022.3163138

Tonyali, S., Cakmak, O., Akkaya, K., Mahmoud, M. M., and Guvenc, I. (2015). Secure data obfuscation scheme to enable privacy-preserving state estimation in smart grid AMI networks. *IEEE Internet Things* 3, 709–719. doi:10.1109/jiot.2015.2510504

Wu, L., Fu, S., Luo, Y., Yan, H., Shi, H., and Xu, M. (2024). A robust and lightweight privacy-preserving data aggregation scheme for smart grid. *IEEE Trans. Dependable Secure Comput.* 21, 270–283. doi:10.1109/tdsc.2023.3252593

Yin, R., Yan, Z., Liang, X., Xie, H., and Wan, Z. (2023). A survey on privacy preservation techniques for blockchain interoperability. *Syst. Archit.*, 140. doi:10.1016/j.sysarc.2023.102892

Yu, H., et al. (2022). Challenges in secure multi-party computation: a systematic review. *Cybersecurity Priv.*

Zhang, X., Huang, C., Zhang, Y., and Cao, S. (2021). Enabling verifiable privacy-preserving multi-type data aggregation in smart grids. *IEEE Trans. Dependable Secure Comput.* 19, 4225–4239. doi:10.1109/tdsc.2021.3124546

Zuo, X., X., Li, L., Peng, H., Luo, S., and Yang, Y. (2020). Privacy-preserving multidimensional data aggregation scheme without trusted authority in smart grid. *IEEE Syst. J.* 15, 395–406. doi:10.1109/JSYST.2020.2994363