IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 27, 2021, accepted February 5, 2022, date of publication February 15, 2022, date of current version March 10, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3151784

Investigation and Application of Differential Privacy in Bitcoin

MERVE CAN KUS[1,2] AND ALBERT LEVI[2], (Senior Member, IEEE)
[1]Research and Development Center, Kuveyt Turk Participation Bank, Cayirova, 41420 Kocaeli, Turkey
[2]Faculty of Engineering and Natural Sciences, Sabanci University, 34956 Istanbul, Turkey

Corresponding author: Merve Can Kus (mervecank@sabanciuniv.edu)

ABSTRACT Bitcoin is one of the best-known cryptocurrencies, which captivated researchers with its innovative blockchain structure. Examinations of this public blockchain resulted in many proposals for improvement in terms of anonymity and privacy. Generally used methods for improvement include mixing protocols, ring signatures, zero-knowledge proofs, homomorphic commitments, and off-chain storage systems. To the best of our knowledge, in the literature, there is no study examining Bitcoin in terms of differential privacy, which is a privacy notion coming up with some mechanisms that enable running useful statistical queries without identifying any personal information. In this paper, we provide a theoretical examination of differential privacy in Bitcoin. Our motivation arises from the idea that the Bitcoin public blockchain structure can benefit from differential privacy mechanisms for improved privacy, both making anonymization and privacy breaches by direct queries impossible, and preserving the checkability of the integrity of the blockchain. We first examine the current Bitcoin implementation for four query functions using the differential privacy formulation. Then, we present the feasibility of the utilization of two differential privacy mechanisms in Bitcoin; the noise addition to the transaction amounts and the user graph perturbation. We show that these mechanisms decrease the fraction of the cases violating differential privacy, therefore they can be used for improving anonymity and privacy in Bitcoin. Moreover, we showcase the noise addition to transaction amounts by using IBM Differential Privacy Library. We compare four differential privacy mechanisms for varying privacy parameter values and determine the feasible mechanisms and the parameters.

INDEX TERMS Anonymity, bitcoin, blockchain, cryptocurrency, differential privacy, graph perturbation, noise addition, privacy.

I. INTRODUCTION

Bitcoin and its blockchain structure proposed in 2008 [1] caused a new era to be opened in digital cash systems with the concept of proof of work and conversion of mining power into money. Since then, although, many blockchain-based digital currencies came out, Bitcoin still remains at the top of the market with $1,171,005,836,167 market capitalization [2] as of November 2021. Since Bitcoin has a public blockchain and transactions are explicitly visible, activities of the users can be tracked and linked, and the user identities can be revealed by linking one of the transactions to off-network information as surveyed in [3] and [4]. For instance, with the knowledge that someone shopped online for 0.000381 BTC from a well-known e-commerce site, Bitcoin addresses that made a 0.000381 BTC valued shopping can be found by querying the Bitcoin address of the site and the transaction amounts equal to 0.000381 from the blockchain. Consequently, room for research came up for anonymity and privacy improvement in Bitcoin, and many academic papers have been published [3]–[7]. In these studies, generally used methods for anonymity and privacy improvement include mixing protocols, ring signatures, zero-knowledge proof, homomorphic commitments, and off-chain storage systems. Some of these studies are implemented, for example, Monero [8] using ring signatures, and Zcash [9] using zero-knowledge proofs are two of the prominent cryptocurrencies.

On the other hand, differential privacy, which was proposed in 2006 [10], [11], is a privacy notion that is related to the distinguishability of the presence/absence of an element in a dataset via query functions. A mechanism is differentially private if this distinguishability is below some threshold.

The associate editor coordinating the review of this manuscript and approving it for publication was Zhe Xiao.

There are methods for providing differential privacy, and these methods can be used for improving privacy. Perturbing data with added noise is a way of providing differential privacy, and this method is used for sharing private data for analysis purposes instead of sharing real data. For instance, in order to ensure differential privacy, data from a health database is shared with researchers under certain rules, e.g., a certain number of queries are allowed, and the actual data is perturbed with the addition of noise. This approach provides *global* differential privacy since the addition of the noise is done after the data aggregation. Differential privacy can be achieved *locally*, as well. In this approach, noise is added before data is aggregated to a database. This local approach is utilized by Apple for collecting data from devices [12] and by Google for collecting data from Chrome web browsers [13]. There is a trade-off between privacy and data utility. Adding more noise improves privacy, but it also decreases data utility. This trade-off is formally controlled using a parameter called epsilon (ε). As ε gets smaller, the amount of noise increases, resulting in improved privacy and decreased utility. There are many studies utilizing differential privacy approaches in different areas, some examples include messaging, health, scheduling for ridesharing, artificial intelligence, deep learning, and software defect prediction [14]–[19].

While researchers are exploring new ways to improve anonymity and privacy in blockchain-based cryptocurrencies, taking extra measures for improving anonymity and privacy complicates checking the integrity of the system. This complication is due to the use of public Bitcoin addresses and transaction amounts to check the integrity of the system. For instance, when the transaction amounts are hidden using a cryptographic approach, the total number of coins in the system cannot be counted, and if someone breaks the system, he can issue coins without being detected. Similarly, when the links between transactions are broken using cryptography, the flow of bitcoins cannot be tracked [3]. Considering these, we hypothesize that the Bitcoin blockchain may benefit from differential privacy, which will not affect the checkability of the integrity of the system. Hiding actual transaction amounts by adding noise can be a way of applying differential privacy. Our motivation for this approach also arises from the fact that perturbing actual data with noise makes anonymization and privacy breaches by direct queries impossible. For instance, in the previously mentioned scenario with 0.000381 BTC valued shopping from a well-known e-commerce site, if some noises are added to the transaction amounts while adding them to the blockchain, a value of 0.000381 would be updated as 0.000383 or 0.000377. Therefore, the detection of these shoppers would be prevented by direct queries. Moreover, there would be no guarantee that the closest value to 0.000381 corresponds to the related transaction.

A. RELATED WORK

To the best of our knowledge, there is no study on the examination of Bitcoin in terms of differential privacy in the literature. There are studies combining differential privacy and blockchain [20]–[24] mostly in general areas. Privacy-preserving solutions for general blockchain structure were studied in [20], and differential privacy was mentioned as a potential solution very briefly. Differential privacy was used in [21] while aggregating crowd data via blockchain by a service provider before sharing it with a data consumer. Differentially private machine learning models via blockchain were studied in [22] and [23]. Differential privacy was used in [24] to obfuscate the results of statistical queries in a differentially private blockchain-based data-sharing model.

The utilization of differential privacy in financial blockchain-based systems for improving anonymity and privacy has recently begun to be considered. Digital currency and international money transfers are considered areas as future applications of differential privacy in blockchain [25]. Correspondingly, inspired by Monero, an approach for a cryptocurrency utilizing differential privacy was introduced [26] as a proposal to Zcash Foundation, and granted; however, there is no follow-up study that details and verifies the approach as of this writing. The addition of noise to transaction amounts in the Ethereum blockchain and analysis according to the Eigen centrality measure was done in [27]. The implementation was done in R using relevant network packages, and January 2019 blockchain transaction data (1,551 transactions) obtained from the Etherscan website [28] was used in the study. A graph structure was formed using these transactions, and the most central nodes were detected before and after adding Gaussian noise to transaction amounts respectively. It was shown that the central nodes changed when the noises were added. The motivation for using centrality comes from the idea that more central nodes are at higher risk of being attacked, therefore, preserving privacy for these nodes is important. In this model, the noise addition is done by dedicated and distributed servers before publishing the transactions online. The actual transaction amounts can be accessed through these servers by authenticated users. The Gaussian parameters were determined trial and error, ε was determined as 0.9, and the delta (δ) was determined as 0.4. This study did not examine other differential privacy mechanisms, nor gave the results for different Gaussian parameter values.

In [29], four variants of differential privacy mechanisms (Laplace, Gaussian, Uniform, and Geometric) were tested in decentralized blockchain-based smart metering. In this system, smart meters act as blockchain nodes sending their real-time data plus noises generated via differential privacy mechanisms to grid utility databases. The grid energy data from [30] was modified accordingly to carry out an experiment for 24-hour usage. The evaluation was carried out on 144 data values ranging between 200 and 1,900. For the implementation, Python libraries NumPy v1.14 and pandas v1.0.3 libraries were used. The Laplace, the Gaussian, and the Geometric mechanisms were compared using different ε values ($\varepsilon = 0.01, 0.05, 0.1, 0.3, 0.7$, and 1), and the same values are used for δ in the Uniform mechanism. The evaluation was done according to Mean Absolute Error (MAE).

MAE is calculated by summing absolute differences between the noisy values and the original readings, and taking the mean. Graphs, showing the original and protected readings, were generated at the stated ε and δ values for the mechanisms. The results showed that the mechanisms provide high privacy by adding a large amount of noise when ε or δ is low ($\varepsilon, \delta = 0.01$), and the privacy reduces gradually as ε or δ increases. Among these four mechanisms, the Geometric and the Laplace are found to be performing better at lower ε values by adding a sharp amount of noise, resulting in higher MAEs. Specifically, the Geometric mechanism is found to be more suitable for protecting high peak values (e.g., high usage), and the Laplace mechanism is found to be more suitable for protecting low peak values (e.g., low usage) at $\varepsilon = 0.01$. It was stated that an adequate amount of noise is added when $\varepsilon, \delta = 0.01$ and 0.05, to protect privacy, and $\varepsilon, \delta = 0.01$ were declared as the most suitable privacy parameters. The MAE values for $\varepsilon, \delta > 0.05$ were not provided in the study.

B. CONTRIBUTION

As far as we are aware, there is no study on the examination of Bitcoin from the differential privacy perspective, and this is the first time bringing together these two. In this paper, first, we examine the current implementation of Bitcoin in terms of differential privacy. Then we examine the utilization of one of the differential privacy approaches, i.e., adding noise to the transaction amounts for four query functions. The noise addition decreases the fraction of the cases violating differential privacy from $1/2$ to $1/4$ for one of the functions, and from $1/4$ to $1/8$ for another function. No decrease can be obtained for two of the functions. Moreover, since the flow between users of Bitcoin can be poured as a graph, inspired by the studies combining differential privacy and graphs [31-34], we examine the applicability of differential privacy mechanisms for graphs to Bitcoin, as well. The graph perturbation decreases the fraction of the cases violating differential privacy from $1/2$ to $1/4$ for three functions, and from $1/4$ to $1/8$ for one of the functions.

Lastly, we demonstrate the practical usage of a differential privacy approach in Bitcoin. We showcase the addition of noise, generated by the differential privacy mechanisms, to the transaction amounts. In this context, we provide brief information on the prominent differential privacy libraries, then we provide the details of Diffprivlib [35], the IBM Differential Privacy Library which we use. Utilizing this library and a sample Bitcoin transaction dataset, we use the Laplace, the Gaussian, the Geometric, and the Uniform mechanisms for noise generation at varying ε and δ values ($\varepsilon = 0.01, 0.05, 0.1, 0.5, 1$ and $\delta = 0.01, 0.05, 0.1, 0.5$), visualize the actual amounts along with the noisy values and evaluate the results according to the MAE. It is observed that the MAEs decrease as ε (or δ) increases, and changing the dataset size, to 100, 1,000, and 10,000, does not make a significant difference in the MAE values. The Laplace mechanism results in the highest MAEs for all dataset sizes and all ε values, providing higher privacy protection compared to the other mechanisms. The Gaussian follows the Laplace, and the Uniform results in the third-highest MAEs. The Geometric mechanism results in the lowest MAEs. Furthermore, we present the results of our examination analyzing the effect of the noise addition for preventing direct queries, i.e., queries for transactions with a specific amount. We introduce a novel metric called *mean ranking offset* (MRO), which gives the average rank change over a dataset after the noise addition when the transactions are sorted by amounts. We use this metric in our experiments for the comparison of the mechanisms and the parameter values. The Laplace mechanism provided the largest MRO values for all ε or δ values considered for a dataset with 100 transactions. The Gaussian follows the Laplace and the Uniform results in the third-highest MROs. The Geometric is found to be ineffective according to the MRO metric, as well. It is observed that the MRO values tend to decrease as ε or δ increases. As a result, the Laplace mechanism is determined as the optimal mechanism for improving anonymity and privacy in Bitcoin within the mechanisms we examined. ε equal or less than 0.5 can be used in the Laplace mechanism for successfully hiding the transaction amounts and ranks.

The organization of this paper is as follows. Section II gives background information about differential privacy, Bitcoin, and blockchain. Section III examines the current implementation of Bitcoin in terms of differential privacy, while Section IV provides the feasibility of the utilization of differential privacy mechanisms in Bitcoin. Section V presents an empirical study on noise addition to transaction amounts. Alternative differential privacy mechanisms are also compared in the section. Section VI gives the summary and the discussion. Finally, we present the conclusion in Section VII.

II. BACKGROUND

A. DIFFERENTIAL PRIVACY

Dwork *et al.* [10] introduced ε-indistinguishability as a new notion of privacy leakage in 2006. A mechanism is defined as ε-indistinguishable if for all databases D_1 and D_2 differing in a single row and for all responses to a query function, the probability of obtaining response r for the database D_1 is within a $(1 + \varepsilon)$ multiplicative factor of the probability of obtaining the same response, r, when the database is D_2. Dwork *et al.* stated that ε-indistinguishability is cording to the Laplace distribution as $\Pr[x] \propto e^{-\varepsilon|x|/S(f)}$ where $S(f)$ is the sensitivity of function f: $D^n \rightarrow \mathbb{R}^d$. $S(f)$ is the smallest number such that for all $D_1, D_2 \in D^n$ which differ in a single row, $\|f(D_1) - f(D_2)\|_1 \leq S(f)$. More noise means more privacy. However, as the amount of noise increases, data utility for analysis decreases, so there is a trade-off between privacy and utility. ε determines the amount of privacy loss, the smaller ε is the better privacy and ε is a parameter chosen by the policy. Dwork *et al.* also called ε as *leakage*.

Differential privacy was defined by Dwork [11] as a new measure in the same year and the formulation of differential privacy is given as follows. A function f is ε-differential private if (1) holds for all datasets D_1 and D_2 which differ

in at most a single row and for all subsets $S \subseteq \text{Range}(f)$ where P denotes probability.

$$P[f(D_1) \in S] \leq exp(\varepsilon) \times P[f(D_2) \in S] \quad (1)$$

For noise calculation, although the Laplace distribution was the first mechanism proposed, the Gaussian [36], the Geometric [37], and the Uniform [38] distributions can be also used for numeric data as an alternative, and the Exponential distribution can be used for non-numeric data [36]. Differential privacy can be applied to graphs, as well. Graph perturbation [39] is the noise graph addition to real graph structure and used for obtaining differentially private graphs [40], [41].

B. BITCOIN AND BLOCKCHAIN

Bitcoin is a distributed, peer-to-peer (P2P) digital currency where no central authority exists. Bitcoin is the unit of the currency, and it is shortened to BTC. Bitcoins can be transferred from one address to another address. A transaction is a transfer of bitcoins. Transaction management and issuance of bitcoins are performed jointly by the peers in the network.

Blockchain is the general ledger of Bitcoin; it is the public record of all transactions, shared between all users and used to verify transactions. Blockchain consists of blocks. A block contains and confirms a part of new waiting transactions. Confirmation means a transaction getting processed by the network and being added to the blockchain. Transactions at each block are hashed, paired, and hashed again until a single hash is obtained, which is the Merkle root [42]. Merkle root is stored in the block header. Each block also includes the hash of the previous block header, which results in a chain of blocks. The basic structure of the blockchain is given in Fig. 1.

Each transaction has at least one input and one output including the address and the amount information. In the input, a user can use bitcoins, which were received as an output in one or more transactions previously. As a result, the flow of bitcoins between transactions also forms a chain structure.

In Bitcoin, everything is transparent; all transactions are publicly announced. The only thing done for anonymity is to keep public keys anonymous, using pseudonyms for the addresses. Everyone can monitor that users transfer bitcoins to each other, but the real names are not provided, only the pseudonyms are used.

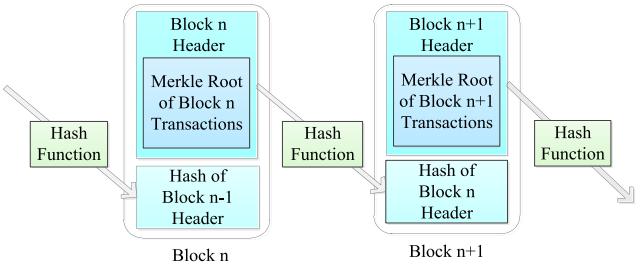

FIGURE 1. The simplified version of the blockchain [3].

III. THEORETICAL EXAMINATION OF BITCOIN FROM DIFFERENTIAL PRIVACY PERSPECTIVE

One can infer that Bitcoin does not provide differential privacy by a pragmatic approach since the presence of a Bitcoin address is explicit in the public Bitcoin blockchain. Although real names are not paired with Bitcoin addresses, addresses can be related to user identities using off-network information [3]. Another argument supporting Bitcoin is not differentially private is the explicitness of transaction amounts and whether a transaction occurred between two specific addresses in the public blockchain. It is worth examining Bitcoin in terms of differential privacy theoretically to confirm these arguments.

The formulation of differential privacy, given as (1), has to be checked to examine Bitcoin in terms of differential privacy theoretically, and finding a counterexample to (1) suffices to detect a violation of differential privacy. In the case of Bitcoin, a set of transactions in the blockchain can be considered as a dataset. In the following subsections, we check the formula for four functions querying; (*i*) transactions between two specific addresses, (*ii*) transactions above a specific amount, (*iii*) transactions for a specific transaction amount, (*iv*) transactions with a specific amount between two specific addresses, as given in Fig. 2. These functions are chosen in the analysis since they can be used for exploiting information from the public blockchain for detecting addresses and deanonymizing users.

A. QUERIES FOR TRANSACTIONS BETWEEN TWO SPECIFIC ADDRESSES

Assume that one wishes to learn whether a transaction occurred between two specific Bitcoin addresses. Let A_1 and A_2 denote the addresses and F be a function that gives the average transaction amount between A_1 and A_2. Let D_1 consists of $n+1$ transactions and D_2 consists of n transactions which are exactly the same as the first n transactions of D_1, which makes D_1 and D_2 differ in a single row. The range of F is between 0 and 21×10^6 BTCs (the maximum number of bitcoins that will ever exist) theoretically. The sensitivity of this function is 21×10^6 divided by the number of transactions in the blockchain. To cover all possible datasets, two cases must be considered; (*i*) the $(n+1)^{st}$ transaction is not a transaction between A_1 and A_2, (*ii*) the $(n+1)^{st}$ transaction is a transaction between A_1 and A_2. The two cases for D_1 and D_2 can be visualized as in Fig. 3. The $(n+1)^{st}$ transaction states, relations between $F(D_1)$ and $F(D_2)$, differential privacy provision or violation statuses in these cases are given in Table 1, where a_{x+1} denotes the $(n+1)^{st}$ transaction amount.

In the first case, $F(D_1)$ equals $F(D_2)$, and the differential privacy formula given in (1) is true for all subsets and ε values. For the second case, $F(D_1)$ equals $F(D_2)$ plus some value that comes from the $(n+1)^{st}$ transaction. The minimum amount that can be

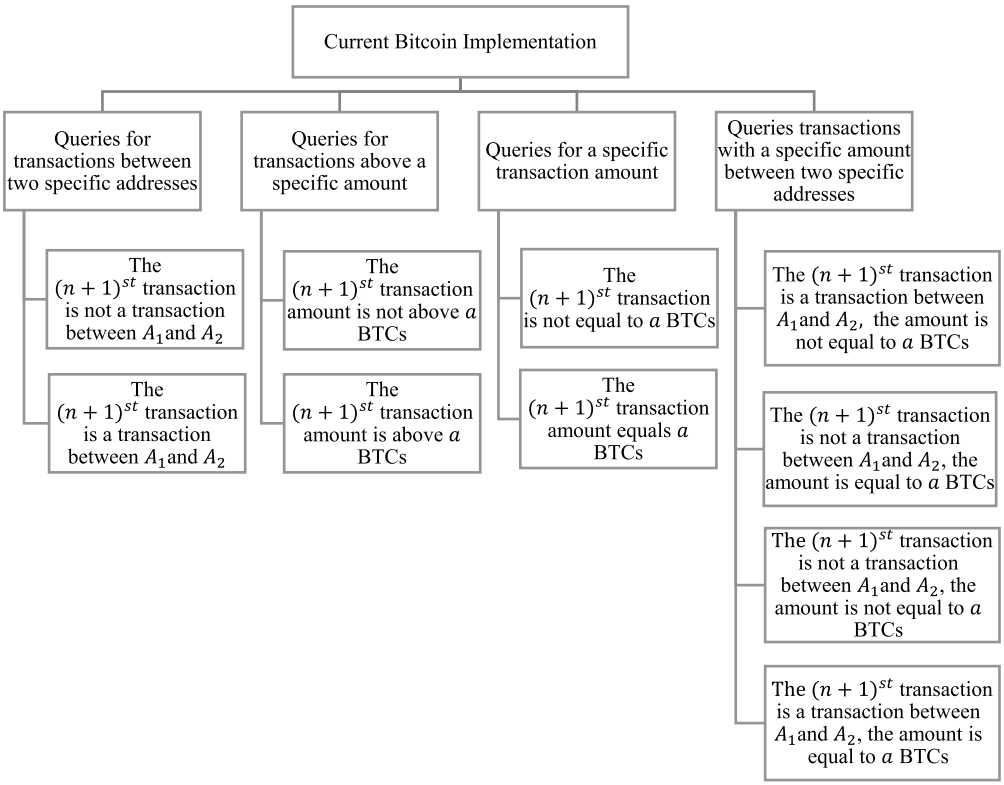

FIGURE 2. Examined cases for the investigation of current Bitcoin implementation from the differential privacy perspective.

transferred in a Bitcoin transaction is 0.00000546. Let S be $\left[F\,(D_2) + \left(0.00000546/\,(n+1)\right), 21x10^6\right]$. The formula (1) turns into (2) with these values.

$$P\left[F\,(D_1) \in [F\,(D_2) + (\frac{0.00000546}{n+1}), 21 \times 10^6]\right] \leq \exp\,(\varepsilon) \times P\left[F\,(D_2) \in [F\,(D_2) + (\frac{0.00000546}{n+1}), 21 \times 10^6]\right] \quad (2)$$

In this formula, $P\left[F\,(D_1) \in \left[F\,(D_2) + (0.00000546/\,(n+1)), 21 \times 10^6\right]\right]$ equals 1, $P\left[F\,(D_2) \in \left[F\,(D_2) + (0.00000546/\,(n+1)), 21 \times 10^6\right]\right]$ equals 0, and (2) turns into (3).

$$1 \leq \exp\,(\varepsilon) \times 0 \quad (3)$$

Since (3) is false for all ε values, this is a violation of differential privacy. This means that there is no differential privacy for a transaction between two Bitcoin addresses in $1/2$ of the cases considered.

B. QUERIES FOR TRANSACTIONS ABOVE A SPECIFIC AMOUNT

As a second examination, assume that one wishes to learn whether a transaction with an amount above a BTCs occurred. Let F be a function that gives the number of transactions having an amount above a BTCs in the blockchain. The sensitivity of this function is 1, since adding a single row to any dataset will change the output by at most 1. Let D_1 consists of $n+1$ transactions and D_2 consists of n transactions that are exactly the same as the first n transactions of D_1. To cover all possible datasets, two cases must be considered; (i) the $(n+1)^{st}$ transaction amount is not above a BTCs, (ii) the $(n+1)^{st}$ transaction amount is above a BTCs. The two cases for D_1 and D_2 can be visualized as in Fig. 4. The $(n+1)^{st}$ transaction states, relations between $F(D_1)$ and $F(D_2)$, differential privacy provision or violation statuses in these cases are given in Table 2.

In the first case, $F\,(D_1)$ equals $F\,(D_2)$, and the differential privacy formula given in (1) is true for all subsets and ε values. For the second case, $F\,(D_1)$ equals $F\,(D_2) + 1$. Consider the case when $F\,(D_2)$ equals 0, i.e., there is no transaction with an amount above a. In this case, $F\,(D_1)$ equals 1. The range of F is $[0, n+1]$ for D_1 and $[0,n]$ for D_2. Let S be $[1,n]$. F is ε-differential private if the following holds.

$$P\,[F\,(D_1) \in [1,n]] \leq \exp\,(\varepsilon) \times P\,[F\,(D_2) \in [1,n]] \quad (4)$$

Since $F\,(D_1)$ equals 1, $P\,[F\,(D_1) \in [1,n]]$ equals 1, and since $F\,(D_2)$ equals 0, $P\,[F\,(D_2) \in [1,n]]$ equals 0, (4) turns into (5), which is false for all ε values, showing a violation of differential privacy. This means that differential privacy is violated for transactions having an amount above a specific value in $1/2$ of the cases considered.

$$1 \leq \exp\,(\varepsilon) \times 0 \quad (5)$$

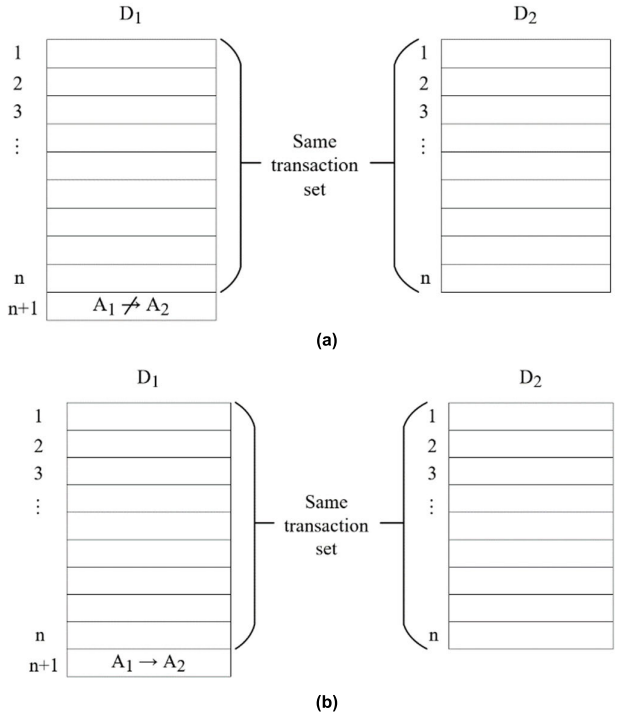

FIGURE 3. Two transaction datasets that differ in a single transaction; (a) The $(n+1)^{st}$ transaction is not a transaction between A_1 and A_2; (b) The $(n+1)^{st}$ transaction is a transaction between A_1 and A_2.

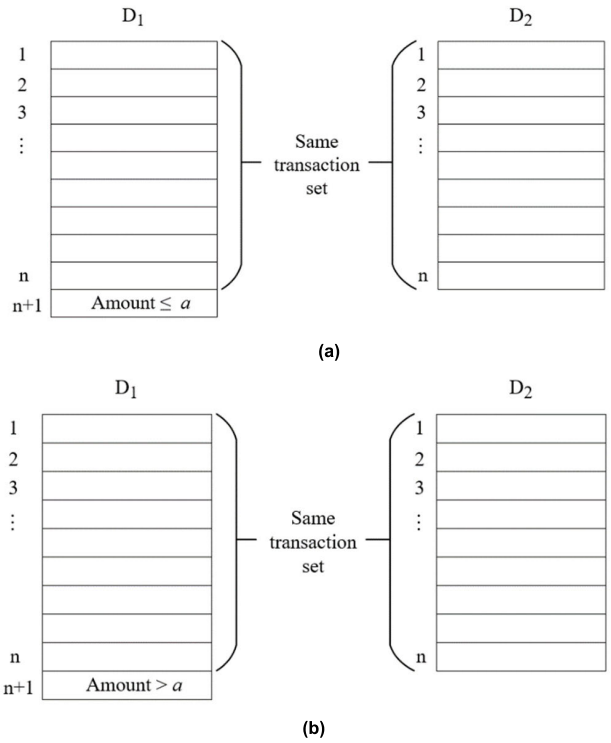

FIGURE 4. Two transaction datasets that differ in a single transaction; (a) The $(n+1)^{st}$ transaction amount is not above a BTCs; (b) The $(n+1)^{st}$ transaction amount is above a BTCs.

TABLE 1. Cases considered in differential privacy evaluation of the queries for transactions between two specific addresses.

Case	$(n+1)^{st}$ Transaction	$F(D_1)$ and $F(D_2)$	Differential Privacy
i	$A_1 \nrightarrow A_2$	$F(D_1) = F(D_2)$	✓
ii	$A_1 \rightarrow A_2$	$F(D_1) = F(D_2) + a_{x+1}/(n+1)$	X

TABLE 2. Cases considered in differential privacy evaluation of the queries for transactions above a specific amount.

Case	$(n+1)^{st}$ Transaction	$F(D_1)$ and $F(D_2)$	Differential Privacy
i	Amount $\leq a$	$F(D_1) = F(D_2)$	✓
ii	Amount > a	$F(D_1) = F(D_2) + 1$	X

C. QUERIES FOR A SPECIFIC AMOUNT

A question that comes to mind might be "What happens if blockchain was sought for transactions with a specific amount?". Pragmatically, it can be said that transactions transferring a specific amount and related Bitcoin addresses can be detected easily from the public blockchain structure. However, a theoretical examination is required to confirm these arguments. Therefore, as the last examination, we evaluate this query theoretically in terms of differential privacy. Assume that one wishes to learn whether there is a transaction with an amount equal to a BTCs. Let F be a function that gives the number of transactions that have an amount equal to a BTCs in the blockchain. The sensitivity of this function is 1, as well. Let D_1 consists of $n+1$ transactions and D_2 consists of n transactions that are exactly the same as the first n transactions of D_1. Again, to cover all possible datasets, two cases must be considered; (i) the $(n+1)^{st}$ transaction amount is not equal to a BTCs, (ii) the $(n+1)^{st}$ transaction amount equals a BTCs. The two cases for D_1 and D_2 can be visualized as in Fig. 5. The $(n+1)^{st}$ transaction states, relations between $F(D_1)$ and $F(D_2)$, differential privacy provision or violation statuses in these cases are given in Table 3.

In the first case, $F(D_1)$ equals $F(D_2)$, and the differential privacy formula given in (1) is true for all subsets and ε values. In the second case, $F(D_1)$ equals $F(D_2)+1$. Consider the case when $F(D_2)$ equals 0, i.e., no transaction amount is equal to a BTCs. In this case, $F(D_1)$ equals 1. The range of F is $[0, n+1]$ for D_1 and $[0,n]$ for D_2. For S is $[1,n]$, there is a violation of differential privacy as shown in the previous query. Again, this means that there is no differential privacy for a specific transaction amount $1/2$ of the cases considered.

D. QUERIES FOR TRANSACTIONS WITH A SPECIFIC AMOUNT BETWEEN TWO SPECIFIC ADDRESSES

Assume that one wishes to learn whether a transaction with an amount a occurred between two specific Bitcoin addresses. Let A_1 and A_2 denote the addresses and F be a function that gives the number of transactions between A_1 and A_2 that have an amount equal to a BTCs. This query function

is basically the combination of the query functions examined in III-A and III-C. Let D_1 consists of $n+1$ transactions and D_2 consists of n transactions which are exactly the same as the first n transactions of D_1, which makes D_1 and D_2 differ in a single row. To cover all possible datasets, four cases must be considered; (*i*) the $(n+1)^{st}$ transaction is a transaction between A_1 and A_2, the amount is not equal to a BTCs, (*ii*) the $(n+1)^{st}$ transaction is not a transaction between A_1 and A_2, the amount is equal to a BTCs, (*iii*) the $(n+1)^{st}$ transaction is not a transaction between A_1 and A_2, the amount is not equal to a BTCs, (*iv*) the $(n+1)^{st}$ transaction is a transaction between A_1 and A_2, the amount is equal to a BTCs. The $(n+1)^{st}$ transaction states, relations between D_1 and D_2, differential privacy provision or violation statuses in these cases are given in Table 4.

In the first three cases, $F(D_1)$ equals $F(D_2)$ since the $(n+1)^{st}$ transaction is not a transaction between A_1 and A_2 that have an amount equal to a BTCs. As a result, the differential privacy formula given in (1) is true for all subsets and ε values for these cases. In the fourth case, $F(D_1)$ equals $F(D_2)+1$. Consider the case when $F(D_2)$ equals 0, i.e., this means that there is no transaction between A_1 and A_2 with an amount a. In this case, $F(D_1)$ equals 1. The range of F is $[0, n+1]$ for D_1 and $[0,n]$ for D_2. For S is $[1,n]$, there is a violation of differential privacy according to the differential privacy formulation. This means that there is no differential privacy for transactions between two specific Bitcoin addresses with a specific amount in $1/4$ of the cases considered.

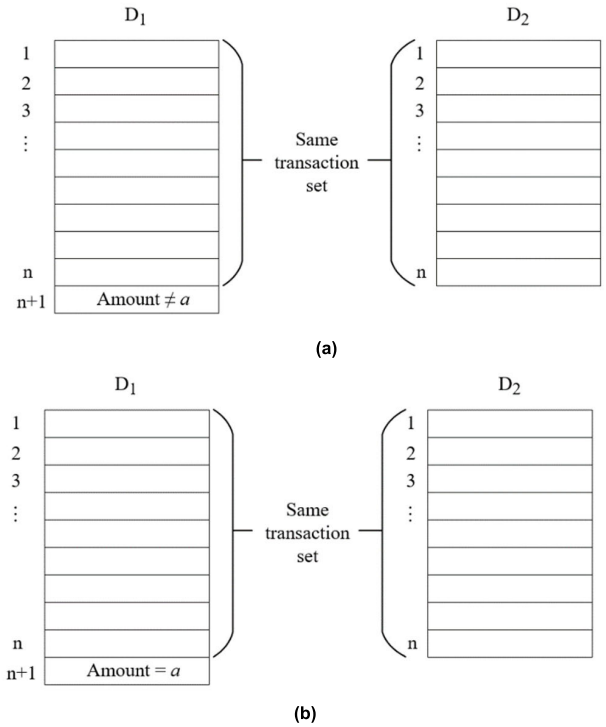

FIGURE 5. Two transaction datasets that differ in a single transaction; (a) The $(n+1)^{st}$ transaction amount is not equal to a BTCs; (b) The $(n+1)^{st}$ transaction amount equals a BTCs.

TABLE 3. Cases considered in differential privacy evaluation of the queries for a specific amount.

Case	$(n+1)^{st}$ Transaction	$F(D_1)$ and $F(D_2)$	Differential Privacy
i	Amount ≠ a	$F(D_1) = F(D_2)$	✓
ii	Amount = a	$F(D_1) = F(D_2)+1$	X

TABLE 4. Cases considered in differential privacy evaluation of the queries for transactions with a specific amount between two specific addresses.

Case	$(n+1)^{st}$ Transaction	D_1 and D_2	Differential Privacy
i	$A_1 \rightarrow A_2$ Amount ≠ a	$F(D_1) = F(D_2)$	✓
ii	$A_1 \nrightarrow A_2$ Amount = a	$F(D_1) = F(D_2)$	✓
iii	$A_1 \nrightarrow A_2$ Amount ≠ a	$F(D_1) = F(D_2)$	✓
iv	$A_1 \rightarrow A_2$ Amount = a	$F(D_1) = F(D_2)+1$	X

IV. FEASIBILITY OF THE UTILIZATION OF DIFFERENTIAL PRIVACY MECHANISMS IN BITCOIN

After examining the current Bitcoin implementation in the previous section, in this section, we investigate the effects of applying differential privacy mechanisms as shown in Fig. 6.

A. NOISE ADDITION TO TRANSACTION AMOUNTS

One way of utilizing differential privacy for improving privacy in Bitcoin may be the addition of Laplace noise to the transaction amounts while including transactions in the blockchain, as a local differential privacy application. This change clearly requires a modification of the Bitcoin transaction verification mechanism, as well. However, this study focuses on the examination of applying differential privacy mechanisms and results in terms of satisfying differential privacy; we leave the actual implementation of such a verification mechanism, and examination of the utility of noise added transaction amounts as a future study. In the following subsections, we examine the effect of noise addition on differential privacy for the four query functions, which were examined in Section III.

1) EFFECT OF NOISE ADDITION ON QUERIES FOR TRANSACTIONS BETWEEN TWO SPECIFIC ADDRESSES

Consider the function in Section III-A provided as an example, where the existence of a transaction between two specific Bitcoin addresses, A_1 and A_2, is sought, and F is a function that gives the average transaction amount between A_1 and A_2. D_1 consists of $n+1$ transactions and D_2 consists of n transactions which are exactly the same as the first n transactions

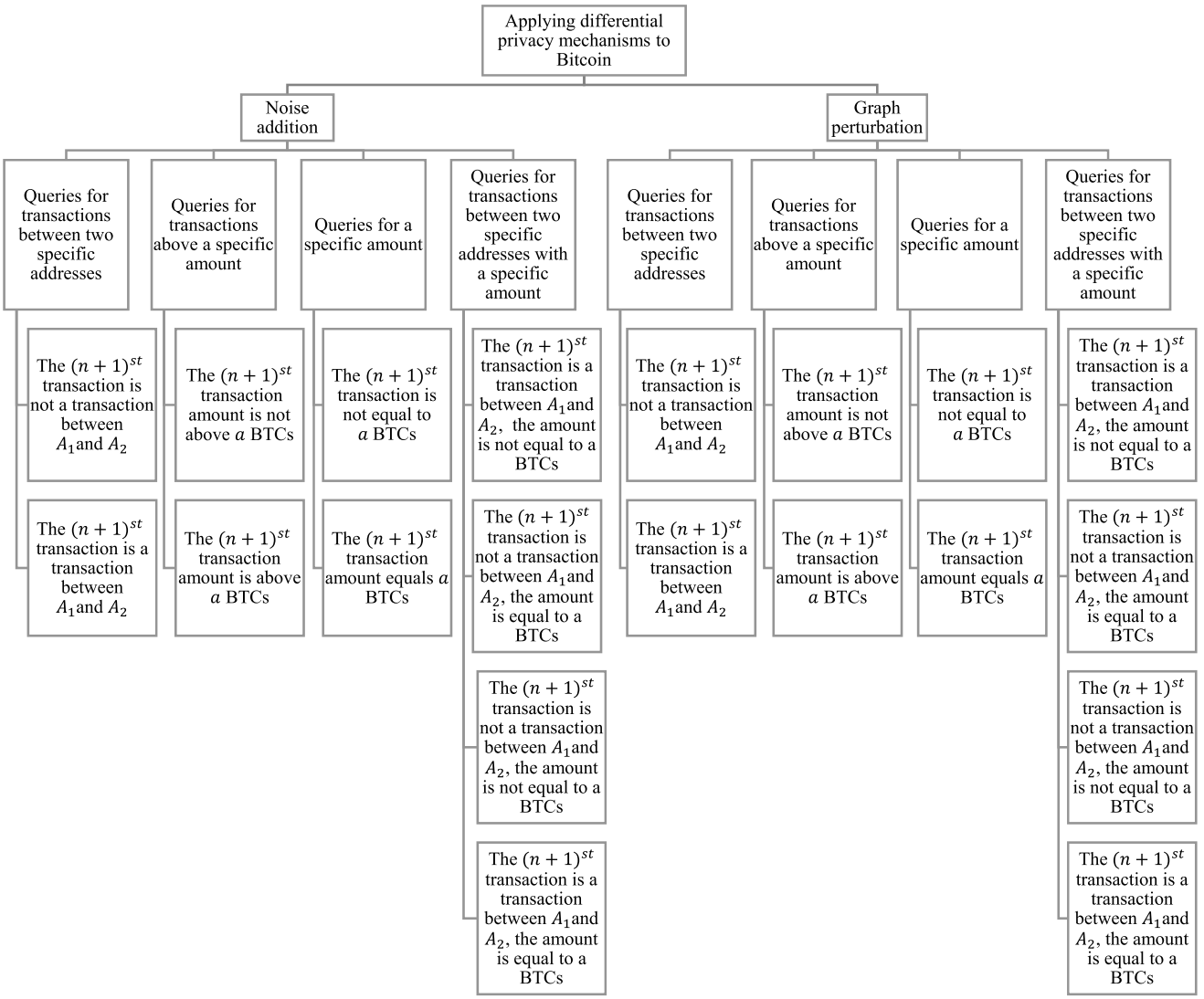

FIGURE 6. Examined cases for the investigation of Bitcoin from the differential privacy perspective with the application of differential privacy mechanisms.

of D_1. Assume that the blockchain stores transactions with noise values generated according to the Laplace mechanism added to the transaction amounts. Moreover, assume that noise values are added accordingly so that the minimum and the maximum Bitcoin transaction amounts do not change, stay as 0.00000546 and 21×10^6 BTCs respectively.

The noise values that will be added can be calculated using the noise distribution function and the sensitivity of the query function. Again, the range of F is between 0 and 21×10^6 BTCs since even in the nonexistence of at least one transaction between A_1 and A_2, the average transaction amount is still 0. To cover all possible datasets, again, two cases must be considered; (i) the $(n+1)^{st}$ transaction is not a transaction between A_1 and A_2, (ii) the $(n+1)^{st}$ transaction is a transaction between A_1 and A_2. The two cases for D_1 and D_2 after the noise addition can be visualized as in Fig. 7. These cases and the corresponding $(n+1)^{st}$ transaction states, relations between $F(D_1)$ and $F(D_2)$, differential privacy provision or violation statuses after the noise addition are given in Table 5, where a_{x+1} denotes the $(n+1)^{st}$ transaction amount.

In the first case, $F(D_1)$ equals $F(D_2)$ after the noise addition, and the differential privacy formula given in (1) is true for all subsets and ε values. For the second case, again, $F(D_1)$ equals $F(D_2)$ plus some value that comes from the noise added $(n+1)^{st}$ transaction. The minimum amount that can be transferred in a Bitcoin transaction is still 0.00000546. For S is $\left[F(D_2) + \left(0.00000546 \big/ (n+1)\right), 21 \times 10^6\right]$, a violation of differential privacy can be shown as in Section III-A in $1/2$ of the cases considered for this query. Thinking pragmatically, it can be inferred that adding noise to transaction amounts does not hide the existence of a transaction between two specific addresses at any level, as well.

TABLE 5. Cases considered in differential privacy evaluation of the queries for transactions between two specific addresses with noise addition.

Case	$(n+1)^{st}$ Transaction	$F(D_1)$ and $F(D_2)$ After Noise	Differential Privacy After Noise
i	$A_1 \nrightarrow A_2$	$F(D_1) = F(D_2)$	✓
ii	$A_1 \rightarrow A_2$	$F(D_1) = F(D_2) + a_{x+1}/(n+1)$	X

TABLE 6. Cases considered in differential privacy evaluation of the queries for transactions above a specific amount with noise addition.

Case	$(n+1)^{st}$ Transaction Before Noise	$(n+1)^{st}$ Transaction After Noise	$F(D_1)$ and $F(D_2)$ After Noise	Differential Privacy After Noise
i	Amount $\leq a$	Amount $\leq a$	$F(D_1) = F(D_2)$	✓
		Amount $> a$	$F(D_1) = F(D_2) + 1$	X
ii	Amount $> a$	Amount $\leq a$	$F(D_1) = F(D_2)$	✓
		Amount $> a$	$F(D_1) = F(D_2) + 1$	X

2) EFFECT OF NOISE ADDITION ON QUERIES FOR TRANSACTIONS ABOVE A SPECIFIC AMOUNT

Consider the function in Section III-B, where one wishes to learn whether a transaction with an amount above a BTCs occurred and F is a function that gives the number of transactions greater than a BTCs in the blockchain. Again, let D_1 consists of $n+1$ transactions, and D_2 consists of n transactions which are exactly the same as the first n transactions of D_1. To cover all possible datasets, two cases must be considered again; (i) the $(n+1)^{st}$ transaction amount is not above a BTCs, (ii) the $(n+1)^{st}$ transaction amount is above a BTCs. The two cases for D_1 and D_2 after the noise addition can be visualized as in Fig. 8. These cases and the corresponding $(n+1)^{st}$ transaction states, relations between $F(D_1)$ and $F(D_2)$, differential privacy provision or violation statuses after the noise addition are given in Table 6.

In the first case, there are two possible outcomes. $F(D_1)$ may be equal to $F(D_2)$ after the noise addition if the amount remains not above a. In this situation, the differential privacy formula given in (1) is true for all subsets and ε values. If the amount gets greater than a, $F(D_1)$ gets equal to $F(D_2)+1$. For the second case, there are two possible outcomes, as well. $F(D_1)$ may be equal to $F(D_2)$ if a negative noise is added to the $(n+1)^{st}$ transaction, which results in a transaction amount below a BTCs and true for the differential privacy formula given in (1) for all subsets and ε values. Alternatively, $F(D_1)$ may be equal to $F(D_2)+1$, if a positive noise is added to the $(n+1)^{st}$ transaction, which results in a violation of differential privacy as shown in Section III-B. The differential privacy is violated for this query in $2/4$ of the cases considered.

3) EFFECT OF NOISE ADDITION ON QUERIES FOR A SPECIFIC TRANSACTION AMOUNT

Consider the function in Section III-C, where one wishes to learn whether a transaction with an amount equal to a BTCs occurred and F is a function that gives the number of transactions with the amount a in the blockchain. Let D_1 and D_2 be two neighbor datasets that consist of exactly the same n transactions and D_1 has an additional $(n+1)^{st}$ transaction. For this query function, two cases must be considered to cover all possible datasets; (i) the $(n+1)^{st}$ transaction amount is not equal to a BTCs, (ii) the $(n+1)^{st}$ transaction amount is equal to a BTCs. The two cases for D_1 and D_2 after the noise addition can be visualized as in Fig. 9. These cases and the corresponding $(n+1)^{st}$ transaction states, relations between $F(D_1)$ and $F(D_2)$, differential privacy provision or violation statuses after the noise addition are given in Table 7.

In the first case, two outcomes can occur after the noise addition; $(i.i)$ $(n+1)^{st}$ transaction amount gets a value different from a BTCs, $(i.ii)$ $(n+1)^{st}$ transaction amount gets equal to a BTCs. In case $(i.i)$, the numbers of transactions having an amount equal to a BTCs are equal for D_1 and D_2, and $F(D_1)$ is equal to $F(D_2)$, therefore, differential privacy is provided. In case $(i.ii)$, $F(D_1)$ equals $F(D_2)+1$. Consider the case when $F(D_2)$ equals 0, i.e., no transaction amount is equal to a BTCs after the noise addition. In this case, $F(D_1)$ is equal to 1. The range of F is $[0, n+1]$ for D_1 and $[0,n]$ for D_2. For S is $[1,n]$, the violation of differential privacy can be shown as in Section III-C. In the second case, when Laplace noise values are added to the amounts in these datasets, $(n+1)^{st}$ transaction of D_1 has no longer an amount equal to a. Remaining n transactions are the same for D_1 and D_2, and when the noise values are added to the amounts, these n transactions again be the same. As a result, $F(D_1)$

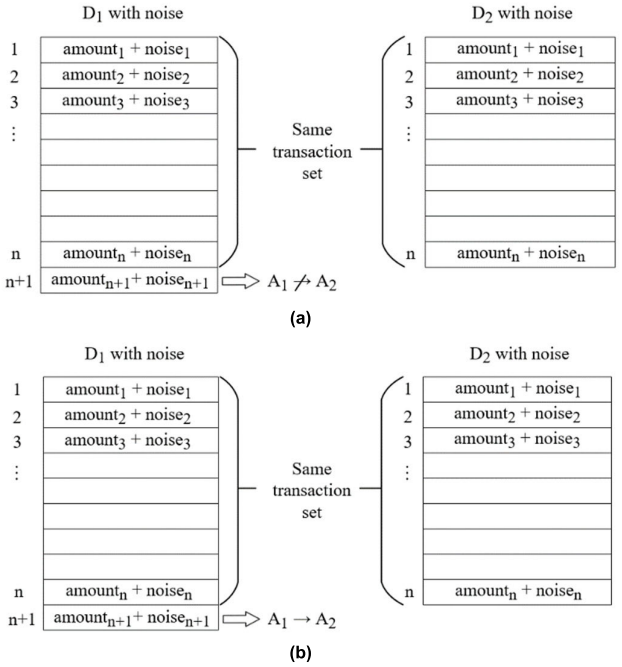

FIGURE 7. Two transaction datasets that differ in a single transaction after the noise addition; (a) The $(n+1)^{st}$ transaction is not a transaction between A_1 and A_2; (b) The $(n+1)^{st}$ transaction is a transaction between A_1 and A_2.

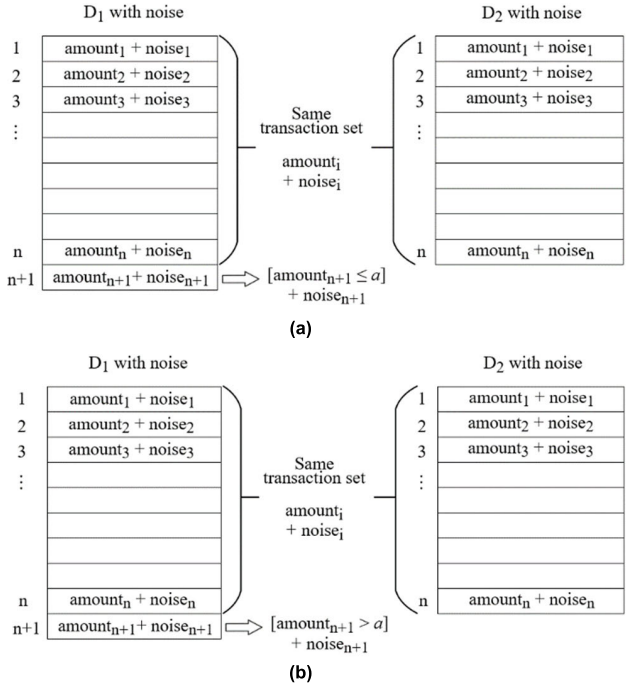

FIGURE 8. Two transaction datasets that differ in a single transaction after the noise addition; (a) The $(n+1)^{st}$ transaction amount is not above a BTCs; (b) The $(n+1)^{st}$ transaction amount is above a BTCs.

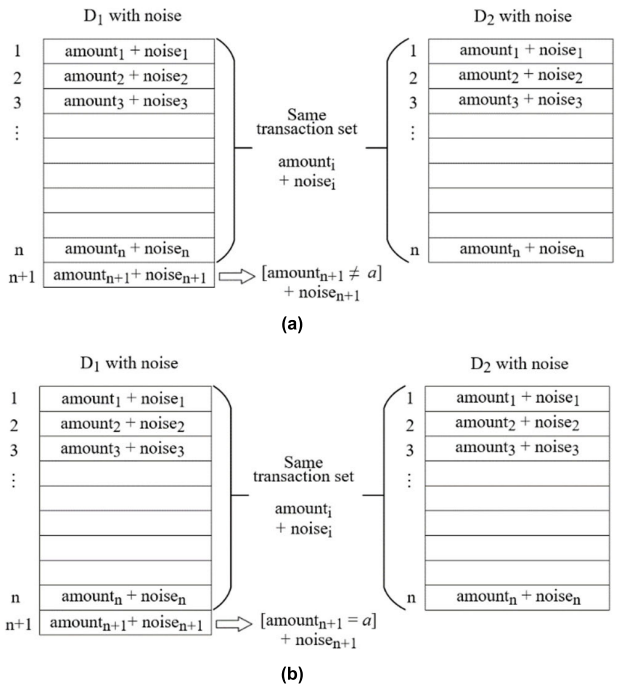

FIGURE 9. Two transaction datasets that differ in a single transaction after the noise addition; (a) The $(n+1)^{st}$ transaction amount is not equal to a BTCs; (b) The $(n+1)^{st}$ transaction amount is equal to a BTCs.

equals $F(D_2)$, and the differential privacy formula given in (1) is true for all subsets and ε values. For case (i), 1/2 of the cases violates differential privacy, and for case (ii), there is no differential privacy violation. For this query, the weighted average of the differential privacy violation becomes $\frac{1}{2} \times \frac{1}{2} + \frac{1}{2} \times 0 = {}^{1}/_{4}$.

4) EFFECT OF NOISE ADDITION ON QUERIES FOR TRANSACTIONS WITH A SPECIFIC AMOUNT BETWEEN TWO SPECIFIC ADDRESSES

Consider the function in Section III-D, where one wishes to learn whether a transaction with an amount equal to a BTCs occurred between two specific Bitcoin addresses. Let A_1 and A_2 denote the addresses and F be a function that gives the number of transactions between A_1 and A_2 that have an amount equal to a BTCs. Let D_1 and D_2 be two neighbor datasets that consist of exactly the same n transactions and D_1 has an additional $(n+1)^{st}$ transaction. For this query function, four cases must be considered to cover all possible datasets; (i) the $(n+1)^{st}$ transaction is a transaction between A_1 and A_2, the amount is not equal to a BTCs, (ii) the $(n+1)^{st}$ transaction is not a transaction between A_1 and A_2, the amount is equal to a BTCs, (iii) the $(n+1)^{st}$ transaction is not a transaction between A_1 and A_2, the amount is not equal to a BTCs, (iv) the $(n+1)^{st}$ transaction is a transaction between A_1 and A_2, the amount is equal to a BTCs. These cases and the corresponding $(n+1)^{st}$ transaction states, relations between $F(D_1)$ and $F(D_2)$, differential privacy provision or violation statuses after the noise addition are given in Table 8.

The range of F is $[0, n+1]$ for D_1 and $[0,n]$ for D_2. When $F(D_1)$ equals $F(D_2)+1$, the violation of differential privacy can be shown by considering the case when $F(D_2)$ equals 0, and $F(D_1)$ is equal to 1 for S is $[1,n]$. When $F(D_1)$ equals $F(D_2)$, the differential privacy formula given in (1) is true for all subsets and ε values. For case (i), 1/2 of the cases violates differential privacy, and for cases $(ii-iv)$, there is no differential privacy violation. For this query, the weighted average of the differential privacy violation becomes $\frac{1}{4} \times \frac{1}{2} + \frac{1}{4} \times 0 + \frac{1}{4} \times 0 + \frac{1}{4} \times 0 = {}^{1}/_{8}$.

B. USER GRAPH PERTURBATION

Another potential way of provisioning differential privacy in Bitcoin is the perturbation of the user graph. In the user graph, also named the user network, the flow of bitcoins between users over time is depicted as a directed graph [3]. Nodes represent users, namely Bitcoin addresses, and directed edges

TABLE 7. Cases considered in differential privacy evaluation of the queries for transactions with a specific amount with noise addition.

Case	$(n+1)^{st}$ Transaction Before Noise	$(n+1)^{st}$ Transaction After Noise	$F(D_1)$ and $F(D_2)$ After Noise	Differential Privacy After Noise
i	Amount $\neq a$	Amount $\neq a$	$F(D_1) = F(D_2)$	✓
		Amount $= a$	$F(D_1) = F(D_2) + 1$	X
ii	Amount $= a$	Amount $\neq a$	$F(D_1) = F(D_2)$	✓

represent the flow of bitcoins between users. An example of the user graph is given in Fig. 10.

Graph perturbation can be applied as adding dummy edges, i.e., dummy transactions, between users or deleting some existing edges, i.e., actual transactions. This change also requires a change of the Bitcoin transaction verification mechanism. Again, our focus in this paper is on the examination of applying differential privacy mechanisms and the corresponding results; we leave the design of such a verification mechanism, and examination of the utility of perturbed transaction graph as future work. In the following subsections, we examine the effect of graph perturbation on differential privacy for the four query functions, which were examined in Section III and Section IV-A.

1) EFFECT OF GRAPH PERTURBATION ON QUERIES FOR TRANSACTIONS BETWEEN TWO SPECIFIC ADDRESSES

Consider the query function that was given in Section III-A, i.e., one wishes to learn whether a transaction occurred between two specific Bitcoin addresses, A_1 and A_2. Let D_1 and D_2 be two neighbor datasets that consist of exactly the same n transactions and D_1 has an additional $(n+1)^{st}$ transaction. Example graphs for D_1 and D_2 are given in Fig. 11.

Let F be a function that gives the average transaction amount between A_1 and A_2. For this query function, two cases must be considered to cover all possible datasets; (i) $(n+1)^{st}$ transaction is between A_1 and A_2, (ii) $(n+1)^{st}$ transaction is not between A_1 and A_2. In the first case, when graph perturbation is applied to these datasets, the following two cases can occur:

- In D_1, the graph perturbation deletes the $(n+1)^{st}$ transaction. Between A_1 and A_2, no or some dummy transactions may be added. In any case, $F(D_1)$ equals $F(D_2)$, and the differential privacy formula given in (1) is true for all subsets and ε values.
- In D_1, the graph perturbation does not delete the $(n+1)^{st}$ transaction. Between A_1 and A_2, no or some dummy transactions may be added. In any case, $F(D_1)$ equals $F(D_2)+1$. When $F(D_2)$ equals 0, for S is $[1,n]$, there is a violation of differential privacy.

In the second case, when graph perturbation is applied to D_1 and D_2, since $(n+1)^{st}$ transaction is not between A_1 and A_2, in the end, $F(D_1)$ equals $F(D_2)$. As a result, the differential privacy formula given in (1) is true for all subsets and ε values.

For case (i), 1/2 of the cases violates differential privacy, and for case (ii), there is no differential privacy violation. For this query, the weighted average of the differential privacy violation becomes $\frac{1}{2}\times\frac{1}{2}+\frac{1}{2}\times 0 = 1/4$.

2) EFFECT OF GRAPH PERTURBATION ON QUERIES ABOVE A SPECIFIC TRANSACTION AMOUNT

Consider the query function that was given in Section III-B, i.e., one wishes to learn whether a transaction with an amount above a BTCs occurred. F is a function that gives the number of transactions greater than a BTCs in the blockchain. D_1 and D_2 are two neighbor datasets as described in the previous query function. Again, two cases must be considered to cover all possible datasets; (i) $(n+1)^{st}$ transaction is above a BTCs, (ii) $(n+1)^{st}$ transaction is not above a BTCs. In the first case, when graph perturbation is applied to these datasets, the following two cases can occur:

- In D_1, the graph perturbation deletes the $(n+1)^{st}$ transaction. No or some dummy transactions above a BTCs may be added. In any case, $F(D_1)$ equals $F(D_2)$, and the differential privacy formula given in (1) is true for all subsets and ε values.
- In D_1, the graph perturbation does not delete the $(n+1)^{st}$ transaction. No or some dummy transactions above a BTCs may be added. In any case, $F(D_1)$ equals $F(D_2)+1$. When $F(D_2)$ equals 0, for S is $[1,n]$, there is a violation of differential privacy.

In the second case, when graph perturbation is applied to D_1 and D_2, since $(n+1)^{st}$ transaction is not above a BTCs, in the end, $F(D_1)$ equals $F(D_2)$. As a result, the differential privacy formula given in (1) is true for all subsets and ε values.

Again, for case (i), 1/2 of the cases violates differential privacy, and for case (ii), there is no differential privacy violation. For this query, the weighted average of the differential privacy violation becomes $\frac{1}{2}\times\frac{1}{2}+\frac{1}{2}\times 0 = 1/4$.

TABLE 8. Cases considered in differential privacy evaluation of the queries for transactions with a specific amount between two specific addresses with noise addition.

Case	$(n+1)^{st}$ Transaction Before Noise	$(n+1)^{st}$ Transaction After Noise	$F(D_1)$ and $F(D_2)$ After Noise	Differential Privacy After Noise
i	$A_1 \rightarrow A_2$ Amount $\neq a$	$A_1 \rightarrow A_2$ Amount $\neq a$	$F(D_1) = F(D_2)$	✓
		$A_1 \rightarrow A_2$ Amount $= a$	$F(D_1) = F(D_2)+1$	X
ii	$A_1 \nrightarrow A_2$ Amount $= a$	$A_1 \nrightarrow A_2$ Amount $\neq a$	$F(D_1) = F(D_2)$	✓
iii	$A_1 \nrightarrow A_2$ Amount $\neq a$	$A_1 \nrightarrow A_2$ Amount $\neq a$	$F(D_1) = F(D_2)$	✓
		$A_1 \nrightarrow A_2$ Amount $= a$	$F(D_1) = F(D_2)$	✓
iv	$A_1 \rightarrow A_2$ Amount $= a$	$A_1 \rightarrow A_2$ Amount $\neq a$	$F(D_1) = F(D_2)$	✓

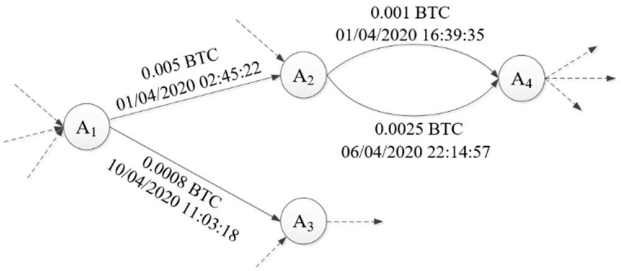

FIGURE 10. A sample Bitcoin user graph.

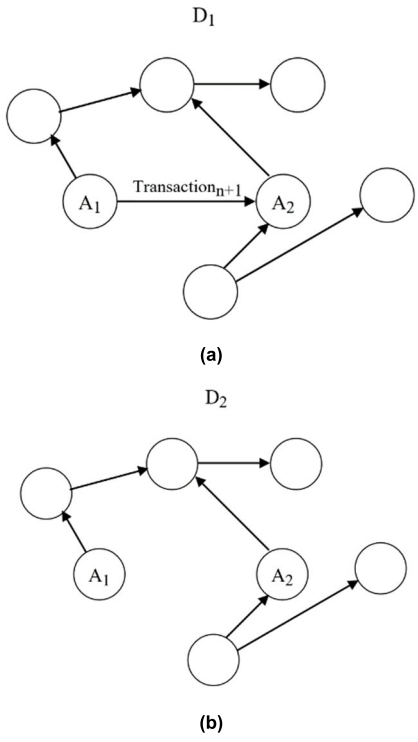

FIGURE 11. (a) D_1 consists of $n+1$ transactions that n of them are exactly the same with the n transactions of D_2 and an $(n+1)^{st}$ transaction which is between A_1 and A_2; (b) D_2 is a dataset that has exactly the same n transactions of D_1.

3) EFFECT OF GRAPH PERTURBATION ON QUERIES FOR A SPECIFIC TRANSACTION AMOUNT

Consider the query function given in Section III-C, i.e., one wishes to learn whether a transaction with an amount equal to a BTCs occurred. Function F gives the number of transactions with an amount a in the blockchain. Let D_1 and D_2 be two neighbor datasets as described earlier. For this query function, two cases that must be considered to cover all possible datasets are as follows; (i) $(n+1)^{st}$ transaction amount is a BTCs, (ii) $(n+1)^{st}$ transaction amount is not a BTCs. In the first case, when the graph perturbation is applied to these datasets, the following two cases can occur:

- In D_1, the graph perturbation deletes the $(n+1)^{st}$ transaction. No or some dummy transactions with an amount equal to a BTCs may be added. In any case, $F(D_1)$ equals $F(D_2)$, and the differential privacy formula given in (1) is true for all subsets and ε values.
- In D_1, the graph perturbation does not delete the $(n+1)^{st}$ transaction. No or some dummy transactions with an amount equal to a BTCs may be added. In any case, $F(D_1)$ equals $F(D_2)+1$. When $F(D_2)$ equals 0, for S is $[1,n]$, there is a violation of differential privacy.

In the second case, when graph perturbation is applied to D_1 and D_2, since $(n+1)^{st}$ transaction is not equal to a BTCs, in the end, $F(D_1)$ equals $F(D_2)$. As a result, the differential privacy formula given in (1) is true for all subsets and ε values.

For case (i), 1/2 of the cases violates differential privacy, and for case (ii), there is no differential privacy violation. For this query, the weighted average of the differential privacy violation becomes $\frac{1}{2}\times\frac{1}{2}+\frac{1}{2}\times 0 = 1/4$, as well.

4) EFFECT OF GRAPH PERTURBATION ON QUERIES FOR TRANSACTIONS WITH A SPECIFIC AMOUNT BETWEEN TWO SPECIFIC ADDRESSES

Consider the query function given in Section III-D, i.e., one wishes to learn whether a transaction with an amount equal to a BTCs occurred between two specific Bitcoin addresses. Let A_1 and A_2 denote the addresses and F be a function that gives the number of transactions between A_1 and A_2 that has an amount equal to a BTCs. Let D_1 and D_2 be two neighbor datasets as described earlier. For this query function, four cases that must be considered to cover all possible datasets are as follows; (i) the $(n+1)^{st}$ transaction is a transaction between A_1 and A_2, the amount is not equal to a BTCs, (ii) the $(n+1)^{st}$ transaction is not a transaction between A_1 and A_2, the amount is equal to a BTCs, (iii) the $(n+1)^{st}$ transaction is not a transaction between A_1 and A_2, the amount is not equal to a BTCs, (iv) the $(n+1)^{st}$ transaction is a transaction between A_1 and A_2, the amount is equal to a BTCs.

In the first three cases, since $(n+1)^{st}$ transaction is not a transaction between A_1 and A_2 with an amount equal to a BTCs, in any case, $F(D_1)$ equals $F(D_2)$ after the graph perturbation . As a result, the differential privacy formula given in (1) is true for all subsets and ε values.

In the fourth case, the following two cases can occur:

- In D_1, the graph perturbation deletes the $(n+1)^{st}$ transaction. No or some dummy transactions with an amount equal to a BTCs may be added. In any case, $F(D_1)$ equals $F(D_2)$, and the differential privacy formula given in (1) is true for all subsets and ε values.
- In D_1, the graph perturbation does not delete the $(n+1)^{st}$ transaction. No or some dummy transactions with an amount equal to a BTCs may be added. In any case, $F(D_1)$ equals $F(D_2)+1$. When $F(D_2)$ equals 0, for S is $[1,n]$, there is a violation of differential privacy.

For cases $(i-iii)$, there is no differential privacy violation. For case (iv), 1/2 of the cases violates differential privacy. As a result, the weighted average of the differential privacy violation for this query becomes $\frac{1}{4}\times 0+\frac{1}{4}\times 0+\frac{1}{4}\times 0+\frac{1}{4}\times\frac{1}{2} = 1/8$.

V. AN EMPIRICAL STUDY ON NOISE ADDITION TO TRANSACTION AMOUNTS

After examining theoretically, we demonstrate a practical utilization of a differential privacy approach in Bitcoin in an empirical way in this section. We add noise to the Bitcoin transaction amounts by applying the Laplace, the Gaussian, the Geometric, and the Uniform mechanisms for the noise generation at different ε values, and evaluate the results.

There are several differential privacy libraries to use. SmartNoise [43], [44], which is a joint study of Microsoft

and Harvard School of Engineering and Applied Sciences, Google's differential privacy library [45], and Diffprivlib [35], [46], the IBM Differential Privacy Library, are the prominent alternatives. The comparison of these libraries according to the variety of differential privacy mechanisms they provide is given in Table 9. The current latest SmartNoise version is v0.2.2 as of this writing and this library offers the Laplace, the Gaussian, and the Geometric mechanisms. The current latest Google library version is v0.0.1 as of this writing and it provides the Laplace and the Gaussian mechanisms. The current latest IBM library DiffPrivlib version is v0.4 as of this writing and this library is the one affording the greatest number of mechanisms for numerical values. The library provides the Laplace, the Gaussian, the Geometric, and the Uniform mechanisms for noise generation in order to achieve a differentially private model. Moreover, according to our evaluation, the documentation of DiffPrivlib is more comprehensible and the usage of the mechanisms is more straightforward, compared to the alternatives. As a result, we selected DiffPrivlib with Python support for our experiments.

The referenced publication and the parameter details of the mechanisms provided by Diffprivlib are summarized in Table 10. Regarding the mechanism parameters, ε can have 1 as the maximum value for the Gaussian mechanism, whereas ε can have higher values than 1 for the Laplace, and the Geometric mechanisms. The Uniform mechanism only uses δ instead of ε, and δ can have a maximum of 0.5. The mechanisms also have a parameter for the sensitivity, which is not stated in the table. We use 1 for the sensitivity parameter for all runs since three out of four query functions that we analyzed in Sections III and IV have sensitivity equal to 1. There are some points to be considered while adding noise to the Bitcoin transaction amounts. The minimum amount of bitcoin that can be sent in a transaction is 546 satoshis, which is equivalent to 0.00000546 BTC. Besides, we assume that the maximum amount of bitcoin that can be sent in a transaction at a certain time is equal to the total amount of bitcoins mined until that time. As of April 2021, we take this maximum value as 18,670,000 [47]. Therefore, the minimum value that a noise added amount can get is 0.00000546 BTC, and the maximum value that a noise added amount can get is 18,670,000 BTC, and the noise values must be added accordingly. Diffprivlib offers folded versions of the Laplace and the Geometric mechanisms. In the folded versions, values outside a predefined range are folded back toward the domain around the closest point within the domain. Since the noisy values must be between 0.00000546 BTC and 18,670,000 BTC in our problem, rather than using the *Laplace* and the *Geometric* classes, we used the *LaplaceFolded* and *GeometricFolded* classes. We set the lower and the upper bounds as 0.00000546 and 18,670,000 respectively in these methods. Although Laplace and LaplaceFolded can be used with real numbers, Geometric and GeometricFolded require an integer input. Therefore, while using GeometricFolded, if an amount is not an integer, we multiplied it with 10^8 to make it an integer value, then applied the *randomise* method to obtain the noisy value and then divided the output by 10^8. Since a folded version for the Gaussian mechanism is not provided in the library, the noise addition trial is done until the noisy value falls within the lower and the upper bounds. Another point to consider is that a noise-added value can have a decimal fraction of up to 8 digits since satoshi is the smallest unit of the currency, which is equal to one hundred millionth of a single bitcoin (0.00000001 BTC). Accordingly, outputs of the randomization methods are rounded to 8 decimal places. We utilized the Python NumPy libraries in our implementation.

We used a published dataset including Bitcoin network transactional metadata [48]. We carried out our experiments by adding noises to *in_btc* fields in this dataset, which are the input amounts of the transactions. We used randomly selected transaction data from 01.01.2014 and 02.01.2014.

In our experiments, first, we analyzed the effect of the dataset size on the behavior of the mechanisms. To this end, while applying the mechanisms, we changed the dataset size to 100, 1,000, and 10,000 respectively. For the evaluation, we used mean absolute error (MAE) values, calculated by summing the absolute differences between the noisy amount values and the actual values, and taking the mean. For the ε parameter of the Laplace, the Gaussian, and the Geometric mechanisms, we used 0.01, 0.05, 0.1, 0.5, and 1. For the δ parameter of the Uniform mechanism, we used 0.01, 0.05, 0.1, and 0.5 since this parameter can have a maximum of 0.5. Although ε can have a value greater than 1 in the Laplace and the Geometric mechanisms, our tests showed that the amount of noise generated is insignificant when this value is greater than 1. As a result, we did not include the results for the greater ε values. We used 1 for δ in the Gaussian mechanism in all runs. The results are given in Fig. 12-14. In the figures, there are no bars for the Uniform mechanism when ε is 1 since it cannot be greater than 0.5.

Fig. 12-14 show that changing the dataset size does not make a significant difference in the MAE values. Apart from the dataset size, the figures show that the MAEs decrease as the ε (or δ) value increases. This outcome is expected since privacy reduces as ε (or δ) increases, and the amount of noise reduces consequently. Moreover, changing the dataset size does not make a difference in the order of the mechanisms. The Laplace mechanism results in the highest MAEs for all dataset sizes and all ε values. The Gaussian is the second by adding approximately the half amount of noise compared to

TABLE 9. The comparison of the differential privacy libraries.

Mechanism	SmartNoise v0.2.2	Google v0.0.1	IBM DiffPrivlib v0.4
Laplace	+	+	+
Gaussian	+	+	+
Geometric	+	–	+
Uniform	–	–	+

the Laplace. The Uniform is the third in the MAE ranking by adding approximately a quarter amount of noise compared to the Laplace mechanism. The Geometric mechanism results in the lowest MAEs, which are significantly lower compared to the other mechanisms. While comparing the mechanisms for the same ε value, it can be said that a higher MAE is better since a higher MAE means that the total amount of noise is higher, resulting in higher privacy protection, as in [29]. Accordingly, the Laplace mechanism is the best for hiding transaction amounts by adding a larger amount of noise. The Gaussian comes next, and the Uniform follows the Gaussian. It is expected that the noisy and the actual amounts are close when the Geometric mechanism is used due to the low noise amounts.

We also visualize 100 actual transaction amounts belonging to 01.01.2014 from the dataset and the corresponding noisy values according to the mechanisms for ε equal to 0.01, 0.05, 0.1, 0.5, and 1 and δ equal to 0.01, 0.05, 0.1, and 0.5 in Fig. 15-19. The average of the actual amounts is 1.409928611, the maximum is 14.96900006, and the minimum is 0.001. From the figures, it is observed that the noisy values deviate a lot from the actual amounts in the Laplace, the Gaussian, and the Uniform mechanisms when ε or δ is smaller than 0.5. The fluctuation of the Laplace mechanism is significant when compared to the other mechanisms. The noisy values in the Geometric mechanism seem to be very close to the actual amounts for all ε values. In these figures, it can be seen that mostly positive amounts of noise are added, i.e., the actual amounts are lower than the noisy amounts mostly. This situation is due to that Bitcoin transaction amounts do not allow so much negative amount of noise since there is a minimum threshold of 0.00000546 BTC, which is the minimum transaction amount. Therefore, the mechanisms continue to generate noise until the noisy amount falls between the minimum and the maximum limits. Especially for the lower ε or δ values, i.e. greater noise amounts, the final noisy value tends to be a greater value than the actual value since the maximum limit, which is assumed as 18,670,000 in this study, is quite large.

One of our aims while considering differential privacy for improving anonymity and privacy in Bitcoin has been preventing privacy breaches via direct queries. In the previously mentioned scenario with 0.000381 BTC valued shopping from a well-known e-commerce site, the transactions with the noisy amounts near 0.000381 in the blockchain may be considered as the candidates while attempting to detect the corresponding transaction. Similarly, an observer may think of using the rank information of the transaction with 0.000381 amount value when all transactions in the dataset are sorted by amounts. The transaction with the same rank or the transactions having ranks close in the noise added dataset may be considered as the candidate transactions corresponding to the transaction sought. In order to examine the differential privacy mechanisms from this aspect, we examined the change in the ranks of specific transactions before and after adding noise. The amount of change shows the performance of the mechanism at hiding the actual rank, and a higher change in the rank makes it difficult for an observer to detect a transaction related to a specific transaction amount.

TABLE 10. The details of the mechanisms provided by Diffprivlib.

Mechanism	Reference in the documentation	Parameters	Input Type
Laplace	[10]	ε: float. Must be in [0, ∞]. δ: float. Must be in [0, 1], default: 0.0.	integer
Gaussian	[36]	ε: float. Must be in (0, 1]. δ: float. Must be in (0, 1].	integer
Geometric	[37]	ε: float. Must be in (0, ∞].	float
Uniform	[49]	δ: float. Must be in (0, 0.5].	integer

In our dataset with 100 amount values, we first checked the ranks of the noisy values corresponding to 14.96900006, which is the maximum of the actual amounts, for varying mechanisms and ε (or δ) values. The results are given in Table 11. We observed that the ranks of the noisy values stay the same when ε is 1 for all mechanisms using ε. It can be said that the mechanisms are unable to hide the rank in this value of ε. The rank of the noisy value does not change for all ε values in the Geometric mechanism. The Laplace mechanism hides the actual rank in $^4/_5$ of the cases, the Gaussian mechanism hides the actual rank in $^3/_5$ of the cases, and the Uniform mechanism hides the actual rank in $^2/_4$ of the cases.

Then, we checked the ranks of the noisy values corresponding to 0.001, which is the minimum of the actual amounts, for varying mechanisms and ε values. The results are given in Table 12. Unlike the previous example value, the Laplace and the Gaussian mechanisms hide the actual rank even when ε is 1. Again, the ranks of the noisy values do not change for all ε values in the Geometric mechanism. It can be seen that the Laplace and the Gaussian mechanisms hide the actual rank in all five ε values, and the Uniform mechanism hides the actual rank in all four δ values.

Finally, we checked the ranks of the noisy values corresponding to the randomly selected 0.41510257 value, which is the 59^{th} in the actual amounts in ascending order, for varying mechanisms and ε (or δ) values. The results are given in Table 13. The rank of the noisy value stays the same for all ε values in the Geometric mechanism. The Laplace and the Gaussian mechanisms are successful at hiding the actual rank in all ε values, and the Uniform mechanism successfully hides the actual rank in all δ values.

In order to generalize this approach to the whole dataset, we define a new metric called *mean ranking offset*. The mean ranking offset (MRO) over a dataset is calculated by taking the average of the absolute differences between the ranks of the actual values in the dataset in ascending order and the ranks of the noisy values in ascending order. As the MRO increases, the distances between the ranks of the noisy values and the actual values increase. Therefore, MRO is an indicator of how successful a mechanism is at hiding the

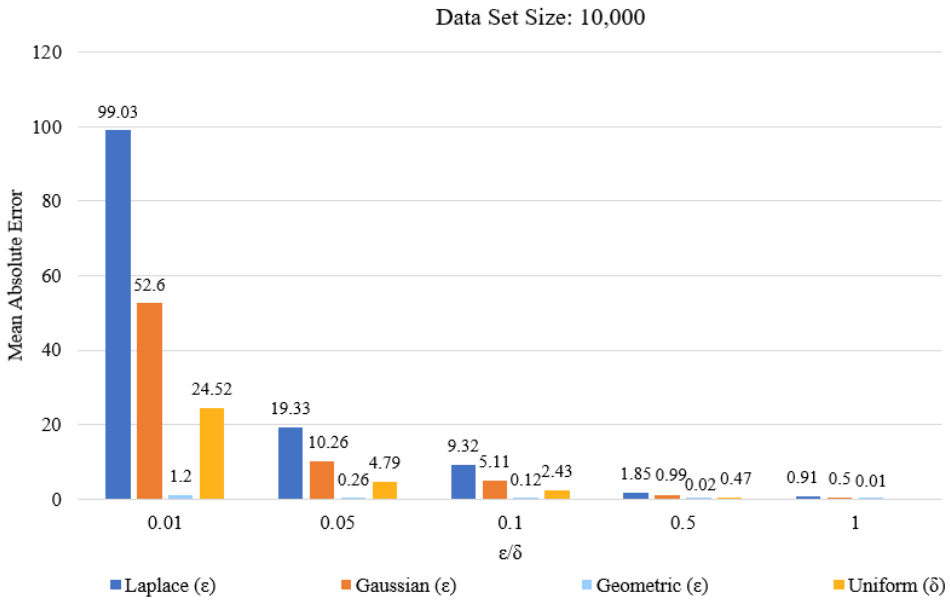

FIGURE 12. Mean absolute errors for varying ε, δ values when the dataset size is 10,000.

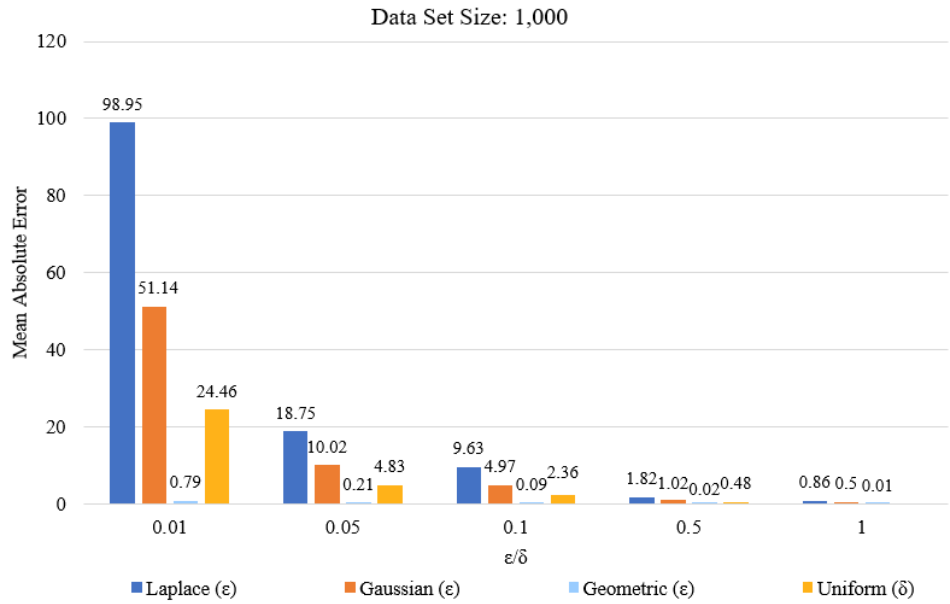

FIGURE 13. Mean absolute errors for varying ε, δ values when the dataset size is 1,000.

actual ranks. We calculated the MRO values over our dataset with 100 transaction amounts for all mechanisms and ε, δ values that we evaluated in the previous analyses. The results are given in Table 14 and visualized in Fig. 20. The largest MRO values are provided by the Laplace mechanism, for all ε (or δ) values considered. It is observed that the mean rank offset values for the Geometric mechanism are very close to 0 and the ineffectiveness of the mechanism compared to the other mechanisms can be clearly seen. For ε, $\delta = 0.01$, the Uniform mechanism follows the Laplace, and the Gaussian mechanism comes after the Uniform. For ε, $\delta = 0.05$, 0.1, and 0.5, the Uniform and the Gaussian change their order, the Gaussian follows the Laplace and the Uniform comes after the Gaussian. δ cannot be 1, therefore MRO is not calculated for the Uniform mechanism in this value. It is observed that as ε or δ increases, MRO values tend to decrease.

VI. SUMMARY AND DISCUSSION

In this section, we summarize our research and observations. In this study, firstly, the current implementation of

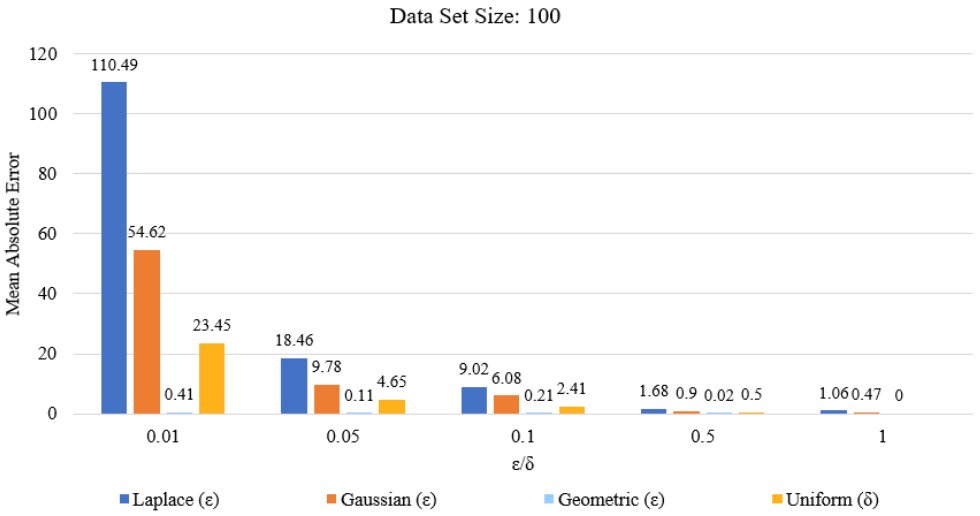

FIGURE 14. Mean absolute errors for varying ε, δ values when the dataset size is 100.

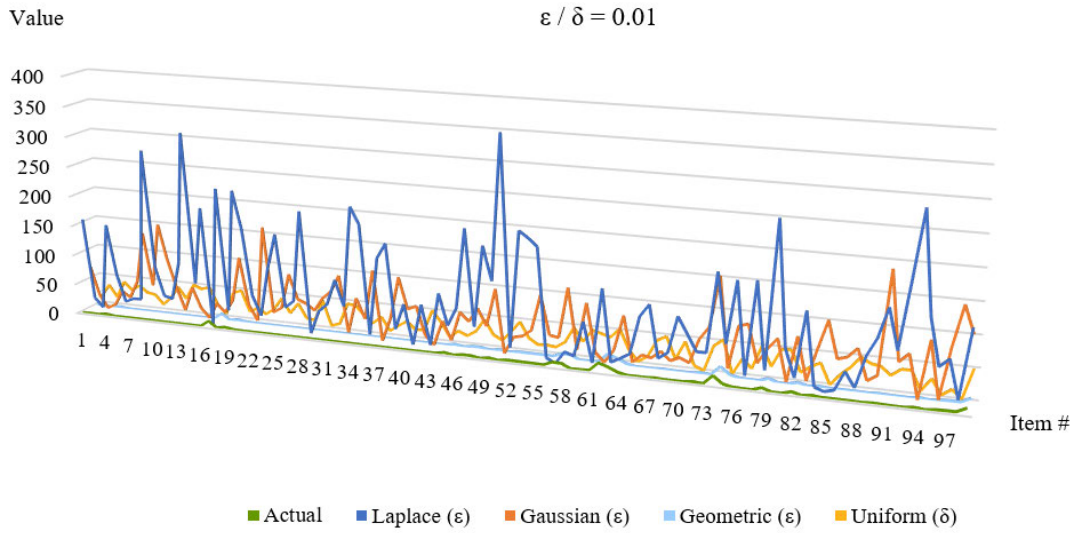

FIGURE 15. The actual transaction amounts along with the noisy amounts when ε or δ is 0.01.

TABLE 11. The rank of the noisy value corresponding to 14.96900006 which is the 1st in the actual amounts in descending order.

	ε or δ				
Mechanism	**0.01**	**0.05**	**0.1**	**0.5**	**1**
Laplace (ε)	18	25	55	6	1
Gaussian (ε)	3	47	6	1	1
Geometric (ε)	1	1	1	1	1
Uniform (δ)	1	3	1	2	N/A

TABLE 12. The rank of the noisy value corresponding to 0.001, which is the 1st in the actual amounts in ascending order.

	ε or δ				
Mechanism	**0.01**	**0.05**	**0.1**	**0.5**	**1**
Laplace (ε)	5	68	77	41	35
Gaussian (ε)	67	24	34	65	14
Geometric (ε)	1	1	1	1	1
Uniform (δ)	32	19	15	49	N/A

Bitcoin is examined for four query functions in terms of differential privacy using the differential privacy formulation. Then, the feasibility of utilizing the noise addition and the graph perturbation mechanisms in Bitcoin is examined for these functions, as well. All possible cases for neighbor datasets are evaluated and the violations are detected.

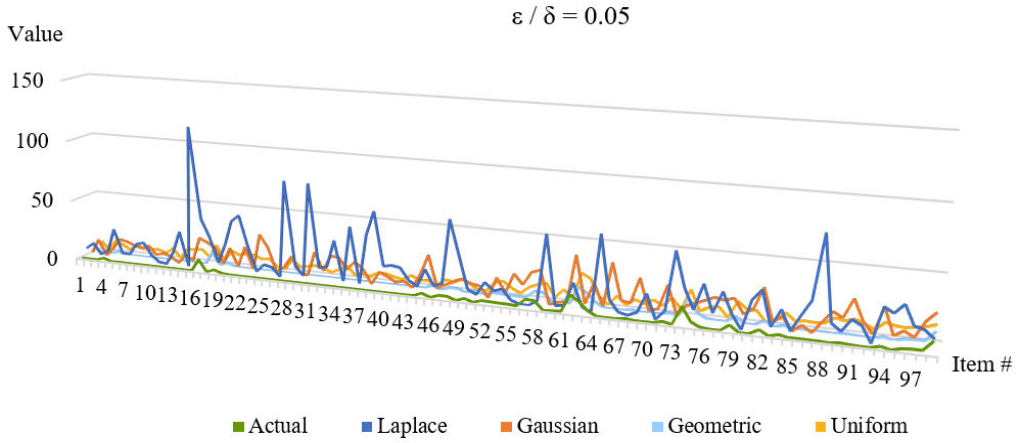

FIGURE 16. The actual transaction along with the noisy amounts when ε or δ is 0.05.

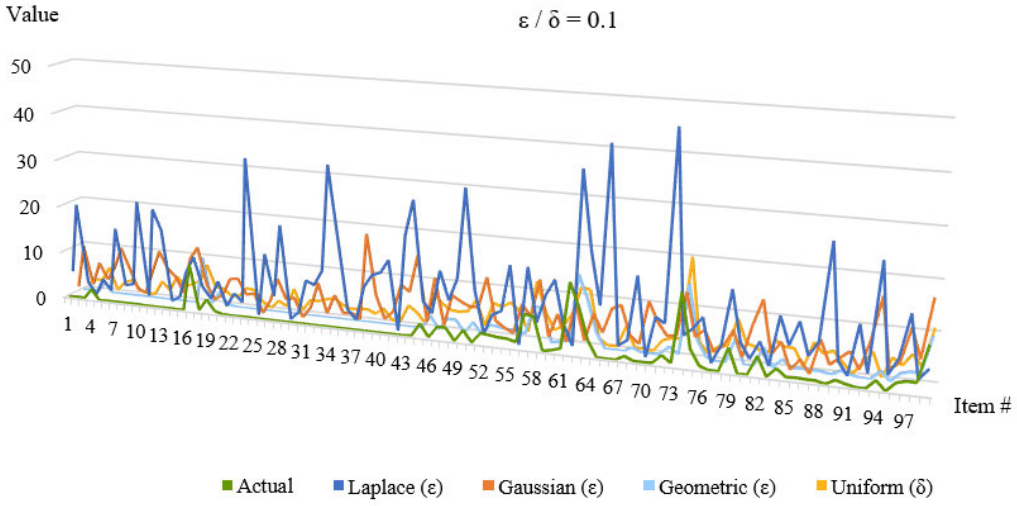

FIGURE 17. The actual transaction amounts along with the noisy amounts when ε or δ is 0.1.

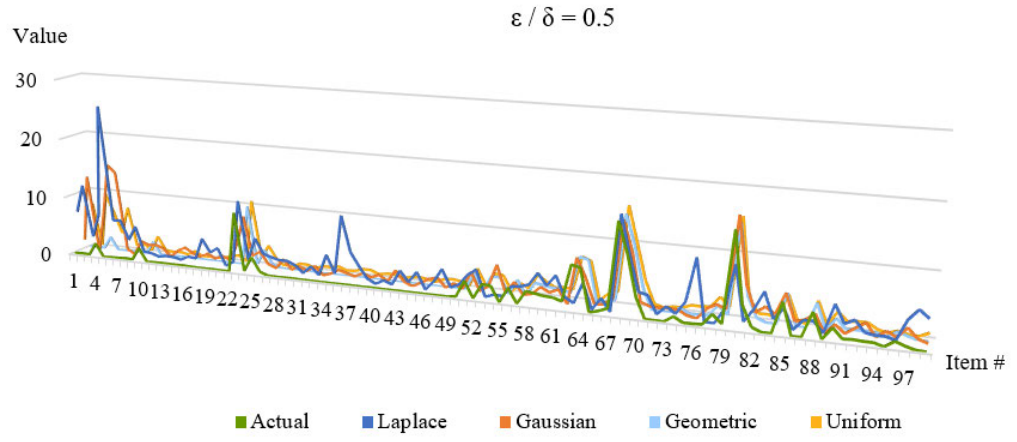

FIGURE 18. The actual transaction amounts along with the noisy amounts when ε or δ is 0.5.

The fractions of the cases violating differential privacy are given in Table 15. The discussed functions query the average transaction amount between two specific addresses, the number of transactions having an amount above a BTCs, and

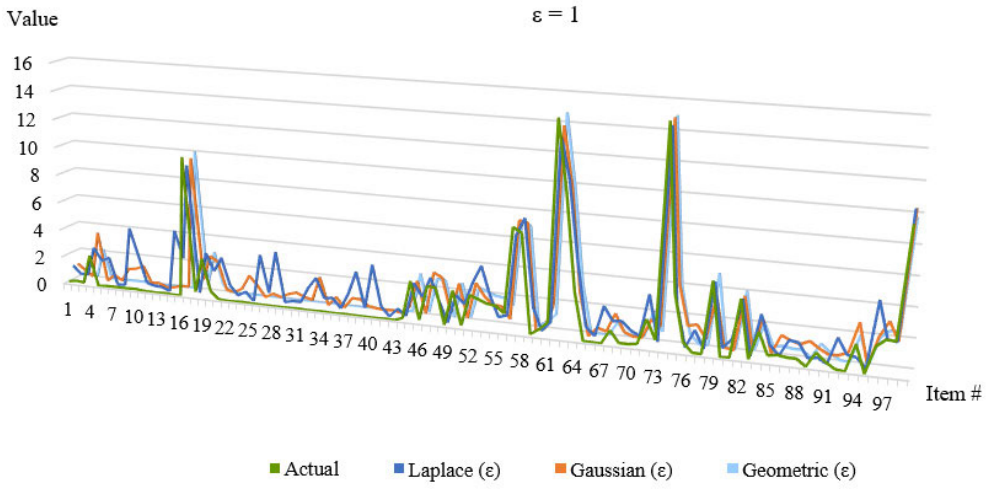

FIGURE 19. The actual transaction amounts along with the noisy amounts when ε is 1.

TABLE 13. The rank of the noisy value corresponding to 0.41510257, which is the 59th in the actual amounts in ascending order.

	ε or δ				
Mechanism	**0.01**	**0.05**	**0.1**	**0.5**	**1**
Laplace (ε)	95	78	39	46	11
Gaussian (ε)	18	71	47	71	56
Geometric (ε)	59	59	59	59	59
Uniform (δ)	78	16	60	65	N/A

TABLE 14. Mean ranking offsets for varying mechanisms and ε/δ values.

	ε or δ				
Mechanism	**0.01**	**0.05**	**0.1**	**0.5**	**1**
Laplace (ε)	35.29	35.07	31.8	27.6	22.06
Gaussian (ε)	31.43	27.76	29.11	23.78	14.38
Geometric (ε)	0.1	0.1	0.06	0.02	0.1
Uniform (δ)	33.07	25.78	25.01	17.84	N/A

the number of transactions having an amount equal to a BTCs, respectively. The selection of these functions was done by considering what an observer would like to learn and get insight from the public blockchain. Interactions between users and the amount values are some meaningful information to use with off-network information.

The current implementation of Bitcoin violates differential privacy in $^1/_2$ of the cases considered for the first three queries and $^1/_4$ of the cases considered for the fourth query. The application of noise addition does not change the fraction of the cases violating differential privacy for the first and the second functions, which query the average transaction amount between two specific addresses, and the

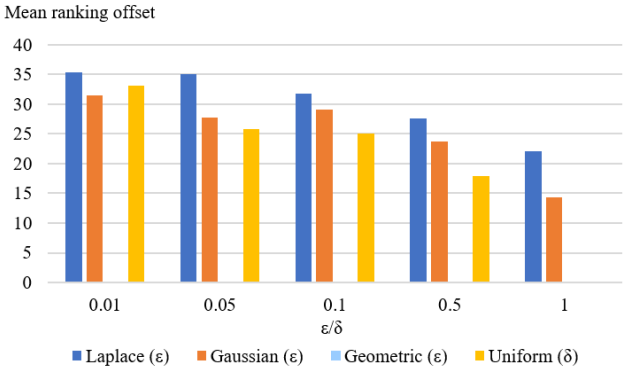

FIGURE 20. Mean ranking offsets for varying mechanisms and ε/δ values.

TABLE 15. The fraction of the cases violating differential privacy.

Query	Function	Current implementation	Noise addition	Graph Perturbation
1	Average transaction amount between A_1 and A_2	$^1/_2$	$^1/_2$	$^1/_4$
2	Number of transactions having an amount above a BTCs	$^1/_2$	$^2/_4$	$^1/_4$
3	Number of transactions having an amount equal to a BTCs	$^1/_2$	$^1/_4$	$^1/_4$
4	Number of transactions between A_1 and A_2 that have an amount equal to a BTCs	$^1/_4$	$^1/_8$	$^1/_8$

number of transactions having an amount above a BTCs, respectively. However, the noise addition decreases the fraction of the cases violating differential privacy to $^1/_4$ for the

third function, which queries the number of transactions having an amount equal to a BTCs. The noise addition decreases the fraction of the cases violating differential privacy to $1/8$ for the fourth function, which queries the number of transactions between two specific addresses with an amount equal to a BTCs.

The graph perturbation decreases the fraction of the cases violating differential privacy to $1/4$ for the first three functions. The fraction of the cases violating differential privacy is decreased to $1/8$ for the fourth function, similar to the noise addition. It can be concluded that both mechanisms can be used to improve anonymity and privacy, whereas the graph perturbation seems to be a better option for the first and the second functions. In these experiments, we covered all possible cases regardless of the amount and exact method of noise addition and perturbation. However, the amount of noise can be calculated using $S(f)$, the sensitivity of a function. For the commonly used Laplace noise mechanism, adding noise with scale $S(f)/\varepsilon$ preserves ε-differential privacy.

Moreover, we demonstrated the utilization of the noise addition to transaction amounts by using the IBM differential privacy library. In our experiments, we examined the Laplace, the Gaussian, the Geometric, and the Uniform mechanisms for generating noise to add to the transaction amount values in a dataset for varying ε and δ values ($\varepsilon = 0.01, 0.05, 0.1, 0.5, 1$, and $\delta = 0.01, 0.05, 0.1, 0.5$). The evaluations are done using MAE values. The results show that the MAEs decrease as ε (or δ) increases, as expected. We observed that the effect of changing the dataset size, to 100, 1,000, and 10,000, does not make a significant difference in the MAE values. The dataset size change also does not make a difference in the order of the mechanisms. We hypothesize that the higher MAE is better since a higher MAE results in higher privacy protection. The Laplace mechanism results in the highest MAEs for all dataset sizes and all ε values. The Gaussian follows the Laplace, and the Uniform results in the third-highest MAEs. The Geometric mechanism is not found effective due to very low MAEs. The behaviors of the mechanisms, in terms of variation, are also noticed when the noisy values generated by the mechanisms for varying ε and δ values are visualized along with the actual amounts for 100 transactions.

We also carried out experiments to analyze the effect of the noise addition on detecting a transaction with a specific amount. We introduced the *mean ranking offset* (MRO) metric, which gives the average rank change over a dataset after the noise addition when the transactions are sorted by amounts. In our evaluation for a dataset with 100 transactions, the Laplace mechanism provided the largest MRO values for all ε or δ values considered. The Gaussian showed a better performance compared to the Uniform in most of the cases and followed the Laplace. The Geometric is ineffective according to the MRO metric, as well. It is observed that the MRO values tend to decrease as ε or δ increases. Moreover, for the maximum and the minimum values in the dataset, we evaluated the mechanisms according to the fraction of the ε or δ values hiding the actual rank. The results are presented in Table 16. It can be seen that the rank of the actual minimum value is successfully hidden for all mechanisms except the Geometric. For hiding the rank of the actual maximum value, there is no mechanism that hides the actual rank for all ε or δ values. However, Laplace performs the best. The Gaussian follows the Laplace, and the Uniform comes after the Gaussian. The Geometric is unsuccessful at hiding both ranks for all mechanisms and ε, δ values.

TABLE 16. The fraction of the ε or δ values hiding the actual rank of the maximum and the minimum values in the dataset.

Mechanism	Maximum	Minimum
Laplace	$4/5$	$5/5$
Gaussian	$3/5$	$5/5$
Geometric	$0/5$	$0/5$
Uniform	$2/4$	$4/4$

As the overall result of our experiments, within the mechanisms and the parameters we examined, the Laplace mechanism can be opted for successfully hiding the transaction amounts and ranks with ε equal or less than 0.5. However, in the previously mentioned related study [29], the most suitable values for ε and δ are determined as 0.01 for generating an adequate amount of noise. This may be due to the range of the values. The values in [29] range between 200 and 1,900, whereas the values used in this study are between 0.001 and 14.96900006 which exemplify the real Bitcoin transaction amounts. Another difference is that the Geometric mechanism is found to be successful for adequate noise generation in [29], whereas our experiments show the opposite by finding this mechanism ineffective.

While attaching the perturbation mechanism to the blockchain, it should be considered that the perturbation should not require a central party since the blockchain is managed collectively by the peers. A reasonable way of perturbation may be triggering and executing the perturbation algorithm automatically while publishing transactions, resulting in perturbed transaction data being added to the blockchain via dedicated and distributed servers as in [27].

Another important point to consider is that the focus of this study was on the examination of applying differential privacy mechanisms and results in terms of satisfying differential privacy. In order to use these differential privacy mechanisms, the verification mechanism must be modified accordingly, and perturbed amounts or transaction graph must be examined in terms of utility. There may be concerns on the effect of the perturbation on the usability of data since hash values used in verification would change, however, these concerns can be addressed with the methods that come from the notion of modifiable blockchains [50], [51] emerged from

the erasing requirements imposed by the GDPR's ''right to be forgotten'' provision.

VII. CONCLUSION

In this study, we present an examination of Bitcoin in terms of differential privacy. Our motivation arises from the fact that differential privacy approaches can be used for improving the privacy of the public Bitcoin blockchain. The differential privacy methods offer the prevention of anonymization and privacy breaches by direct queries and the preservation of checkability of the integrity of the blockchain. We first examine the current Bitcoin implementation using the differential privacy formulation. Then, we examine the application of noise addition to transaction amounts and user graph perturbation as differential privacy mechanisms. Furthermore, we demonstrate an empirical study for practical utilization of the noise addition approach and compare four differential privacy mechanisms according to mean absolute error for varying ε and δ values. In addition, we introduce a new metric called *mean ranking offset* and use it for the comparison, as well. In Section VII, we summarize our observations. It is observed that the noise addition and the graph perturbation mechanisms decrease the fraction of the cases violating differential privacy, therefore they can be used for improving anonymity and privacy in Bitcoin. The noise addition method decreases the fraction of the cases violating differential privacy by half for the three query functions, whereas the graph perturbation method decreases the fraction of the cases violating differential privacy by half for all of the four query functions considered. When the differential privacy mechanisms are compared practically for the noise addition, it is demonstrated that the Geometric mechanism adds a marginal amount of noise for all considered ε values and this mechanism is ineffective at hiding the ranks of the amounts in the dataset. This allows an observer, searching for a transaction with a specific amount, to detect the transaction by finding the nearest noisy value even if the noises are added. Our experiments show that the Laplace mechanism outperforms other mechanisms with high MAE and MRO values, and it can be opted with ε equal or less than 0.5 for improving differential privacy in Bitcoin. Although the results that are obtained in this paper are promising, none of the proposed methods achieved perfect differential privacy. Moreover, there is room for more research. Further research topics include the modification of the verification mechanism accordingly, and examining the effect of the perturbation on the degradation of utility. Moreover, applying these differential privacy mechanisms to other blockchain-based cryptocurrencies may be investigated, as well.

REFERENCES

[1] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[2] *All Cryptocurrencies, CoinMarketCap*. Accessed: Nov. 7, 2021. [Online]. Available: https://coinmarketcap.com/all/views/all

[3] M. C. K. Khalilov and A. Levi, ''A survey on anonymity and privacy in Bitcoin-like digital cash systems,'' *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2543–2585, 3rd Quart., 2018.

[4] M. Conti, E. S. Kumar, C. Lal, and S. Ruj, ''A survey on security and privacy issues of Bitcoin,'' in *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3416–3452.

[5] N. Amarasinghe, X. Boyen, and M. McKague, ''A survey of anonymity of cryptocurrencies,'' in *Proc. Australas. Comput. Sci. Week Multiconf. (ACSW)*, Sydney, NSW, Australia, no. 2, Jan. 2019, pp. 1–10.

[6] S. B. Venkatakrishnan, G. Fanti, and P. Viswanath, ''Dandelion: Redesigning the Bitcoin network for anonymity,'' in *Proc. ACM Meas. Anal. Comput. Syst.*, Urbana-Champaign, IL, USA, no. 22, Jun. 2017, pp. 1–34.

[7] L. H. Zhu, B. K. Zheng, M. Shen, F. Gao, H. Y. Li, and K. X. K. Shi, ''Data security and privacy in Bitcoin system: A survey,'' *J. Comput. Sci. Technol.*, vol. 35, pp. 843–862, 2020.

[8] *Monero—Secure, Private, Untraceable*. Accessed: Feb. 20, 2020. [Online]. Available: https://www.getmonero.org

[9] *Privacy-Protecting Digital Currency | Zcash*. Accessed: Feb. 20, 2020. [Online]. Available: https://z.cash/

[10] C. Dwork, F. McSherry, K. Nissim, and A. Smith, ''Calibrating noise to sensitivity in private data analysis,'' in *Proc. 3rd Conf. Theory Cryptogr. (TCC)*, New York, NY, USA, Mar. 2006, pp. 265–284.

[11] C. Dwork, ''Differential privacy,'' in *Proc. ICALP, Automata, Lang. Program.*, Venice, Italy, Jul. 2006, pp. 1–12.

[12] *Differential Privacy Overview, Apple*. Accessed: Feb. 23, 2020. [Online]. Available: https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf

[13] A. Bittau *et al.*, ''PROCHLO: Strong privacy for analytics in the crowd,'' in *Proc. 26th Symp. Oper. Syst. Princ. (SOSP)*, Shanghai, China, Oct. 2017, pp. 441–459.

[14] D. Lazar, Y. Gilad, and N. Zeldovich, ''Karaoke: Distributed private messaging immune to passive traffic analysis,'' in *Proc. 13th USENIX Conf. Oper. Syst. Design Implement.*, Carlsbad, CA, USA, Oct. 2018, pp. 711–725.

[15] F. K. Dankar and K. E. Emam, ''The application of differential privacy to health data,'' in *Proc. Joint EDBT/ICDT Workshops*, Berlin, Germany, Mar. 2012, pp. 158–166.

[16] W. Tong, J. Hua, and S. Zhong, ''A jointly differentially private scheduling protocol for ridesharing services,'' *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 10, pp. 2444–2456, Oct. 2017.

[17] T. Zhu and P. S. Yu, ''Applying differential privacy mechanism in artificial intelligence,'' in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Dallas, TX, USA, Jul. 2019, pp. 1601–1609.

[18] X. Chen, D. Zhang, Z. Q. Cui, Q. Gu, and X. L. X. Ju, ''DP-Share: Privacy-preserving software defect prediction model sharing through differential privacy,'' *J. Comput. Sci. Technol.*, vol. 34, pp. 1020–1038, 2019.

[19] M. Abadi *et al.*, ''Deep learning with differential privacy,'' in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Vienna, Austria, Oct. 2016, pp. 308–318.

[20] J. B. Bernabe, ''Privacy-preserving solutions for blockchain: Review and challenges,'' *IEEE Access*, vol. 7, pp. 164908–164940, 2019.

[21] H. Duan *et al.*, ''Aggregating crowd wisdom via blockchain: A private, correct, and robust realization,'' in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, Kyoto, Japan, Mar. 2019, pp. 1–10.

[22] X. Chen *et al.*, ''When machine learning meets blockchain: A decentralized, privacy-preserving and secure design,'' in *Proc. IEEE Int. Conf. Big Data*, Seattle, WA, USA, Dec. 2018, pp. 1178–1187.

[23] N. Hynes *et al.*, ''A demonstration of Sterling: A privacy-preserving data marketplace,'' in *Proc. VLDB Endowment*, Rio de Janeiro, Brazil, Aug. 2018, vol. 11, no. 12, pp. 2086–2089.

[24] M. Yang *et al.*, ''Differentially private data sharing in a cloud federation with blockchain,'' *IEEE Cloud Comput.*, vol. 5, no. 6, pp. 69–79, Nov. 2018.

[25] M. Ul Hassan, M. H. Rehmani, and J. Chen, ''Differential privacy in blockchain technology: A futuristic approach,'' *J. Parallel Distrib. Comput.*, to be published, doi: 10.1016/J.JPDC.2020.06.003.

[26] Zcash Foundation, GitHub. *An Alternative Approach to Analyzing Anonymity in Cryptocurrencies*. Accessed: Jan. 11, 2020. [Online]. Available: https://github.com/ZcashFoundation/GrantProposals-2018Q2/issues/36

[27] E. S. Kumar, ''Preserving privacy in Ethereum blockchain,'' *Ann. Data Sci.*, 2020, doi: 10.1007/s40745-020-00279-9.

[28] *Etherscan*. Accessed: Jan. 20, 2021. [Online]. Available: https://etherscan.io

[29] M. U. Hassan, M. H. Rehmani, and J. Chen, ''Performance evaluation of differential privacy mechanisms in blockchain based smart metering,'' Jul. 2020, *arXiv:2007.09802*.

[30] M. Muratori, "Impact of uncoordinated plug-in electric vehicle charging on residential power demand," *Nature Energy*, vol. 3, no. 3, pp. 193–201, Jan. 2018.
[31] S. P. Kasiviswanathan *et al.*, "Analyzing graphs with node differential privacy," in *Proc. 10th Theory Cryptogr. Conf. (TCC)*, Tokyo, Japan, Mar. 2013, pp. 457–476.
[32] Y. Yin, Q. Liao, Y. Liu, and R. Xu, "Structural-based graph publishing under differential privacy," in *Proc. Int. Conf. Cogn. Comput. (ICCC)*, San Diego, CA, USA, Jun. 2019, pp. 67–78.
[33] B. Nguyen *et al.*, "Publishing graph data with subgraph differential privacy," in *Proc. Int. Workshop Inf. Secur. Appl. (WISA)*, Jeju-do, South Korea, Aug. 2015, pp. 134–145.
[34] H. H. Nguyen, A. Imine, and M. Rusinowitch, "Network structure release under differential privacy," *Trans. Data Privacy*, vol. 9, no. 3, pp. 215–241, Dec. 2016.
[35] *IBM Differential Privacy Library—Diffprivlib*. Accessed: Feb. 23, 2021. [Online]. Available: https://diffprivlib.readthedocs.io
[36] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, Aug. 2014.
[37] A. Ghosh, T. Roughgarden, and M. Sundararajan, "Universally utility-maximizing privacy mechanisms," Mar. 2009, *arXiv:0811.2841*.
[38] Q. Geng and P. Viswanath, "Optimal noise adding mechanisms for approximate differential privacy," May 2013, *arXiv:1305.1330*.
[39] V. Torra and J. Salas, "Graph perturbation as noise graph addition: A new perspective for graph anonymization," in *Proc. Data Privacy Manage. (DPM), Cryptocurrencies Blockchain Technol. (CBT)*, Luxembourg, Sep. 2019, pp. 121–137.
[40] A. Sala *et al.*, "Sharing graphs using differentially private graph models," in *Proc. ACM SIGCOMM Conf. Internet Meas.*, Nov. 2011, pp. 81–98.
[41] Z. Jorgensen, T. Yu, and G. Cormode, "Publishing attributed social graphs with formal privacy guarantees," in *Proc. SIGMOD, Int. Conf. Manage. Data*, San Francisco, CA, USA, Jun. 2016, pp. 107–122.
[42] R. C. Merkle, "A digital signature based on a conventional encryption function," in *Advances in Cryptology—CRYPTO* (Lecture Notes in Computer Science), vol. 293. Springer, 1988, pp. 369–378.
[43] *SmartNoise*. Accessed: Jan. 6, 2021. [Online]. Available: https://smartnoise.org
[44] *GitHub—OpenDP/Smartnoise-Core*. Accessed: Jan. 6, 2021. [Online]. Available: https://github.com/opendp/smartnoise-core
[45] *Google's Differential Privacy Libraries*. Accessed: Jan. 6, 2021. [Online]. Available: https://github.com/google/differential-privacy
[46] N. Holohan, S. Braghin, P. M. Aonghusa, and K. Levacher, "Diffprivlib: The IBM differential privacy library," Jul. 2019, *arXiv:1907.02444*.
[47] *Blockchain Charts*. Accessed: Apr. 23, 2021. [Online]. Available: https://www.blockchain.com/charts/total-bitcoins
[48] IEEE DataPort. *Bitcoin Transactions Data 2011–2013*. Accessed: Feb. 3, 2021. [Online]. Available: https://ieee-dataport.org/open-access/bitcoin-transactions-data-2011-2013
[49] Q. Geng, W. Ding, R. Guo, and S. Kumar, "Privacy and utility tradeoff in approximate differential privacy," Feb. 2019, *arXiv:1810.00877*.
[50] E. Politou, F. Casino, E. Alepis, and C. Patsakis, "Blockchain mutability: Challenges and proposed solutions," *IEEE Trans. Emerg. Topics Comput.*, vol. 8, pp. 4737–4744, Oct. 2019.
[51] N. Lee, J. Yang, M. M. H. Onik, and C. Kim, "Modifiable public blockchains using truncated hashing and sidechains," *IEEE Access*, vol. 7, pp. 173571–173582, Nov. 2019.

MERVE CAN KUS received the B.S. and M.S. degrees in computer engineering from Istanbul Technical University, Istanbul, Turkey, in 2006 and 2008, respectively. She is currently pursuing the Ph.D. degree in computer science and engineering with Sabanci University, Istanbul.

She was an Enterprise Architect, the Change Manager, and the FinTech/DisruptiveTech Research and Development Team Leader. She is currently an IT Project Manager. She has been with the Research and Development Center, Kuveyt Turk Participation Bank, Kocaeli, Turkey, since 2010. She is the coauthor of "A Survey on Anonymity and Privacy in Bitcoin-like Digital Cash Systems" which is published in IEEE Communications Surveys and Tutorials, in 2018. Her research interests include blockchain, information security, privacy, digital cash mechanisms, and financial technologies.

Ms. Kus's awards and honors include the Chamber of Electrical Engineers of Turkey Project Competition Honorable Mention, the Ord. Prof. Bedri Karafakioğlu Award, and the Siemens Excellence Award in 2006.

ALBERT LEVI (Senior Member, IEEE) received the Ph.D. degree in computer engineering from Bogazici University, Istanbul, Turkey, in 1999.

He was a Visiting Faculty Member with the Department of Electrical and Computer Engineering, Oregon State University, and a Visiting Professor with the Faculty of Computer Science, Dalhousie University. He is a Professor of computer science and engineering with the Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul, Turkey. He has authored and coauthored more than 100 articles in refereed journals and conferences. His research interests include computer and network security with an emphasis on mobile and wireless system security, public key infrastructures (PKI), privacy, and application layer security protocols.

Dr. Levi was a recipient of the 2018 Turkish Informatics Association Prof. Dr. Aydın Köksal Science Award. He has served in the program committees of various international conferences. He also served as the General and Program Co-Chair of ISCIS 2006, the General Chair of SecureComm 2008, the Technical Program Co-Chair of NTMS 2009, the Publicity Chair of GameSec 2010, and the Program Co-Chair of ISCIS 2011. He is an Editorial Board Member of *The Computer Journal* (Oxford University Press), *Computer Networks* (Elsevier), and *Wireless Networks* (Springer).