

# Privacy-preserving reservation model for public facilities based on public Blockchain

Akbari Indra Basuki<sup>1</sup>, Didi Rosiyadi<sup>1</sup>, Hadi Susanto<sup>1,2</sup>, Iwan Setiawan<sup>1</sup>, Taufik Ibnu Salim<sup>3</sup>

<sup>1</sup>Research Center for Artificial Intelligence and Cyber Security, National Research and Innovation Agency, Bandung, Indonesia

<sup>2</sup>Department of Industrial Engineering, Telkom University, Bandung, Indonesia

<sup>3</sup>Research Center for Smart Mechatronics, National Research and Innovation Agency, Bandung, Indonesia

## Article Info

### Article history:

Received Feb 20, 2024

Revised Mar 21, 2024

Accepted Apr 1, 2024

### Keywords:

Binary masking

Blockchain

Fully homomorphic encryption

Privacy preserving

Public facilities reservation

Reservation table

Smart contract

## ABSTRACT

Ensuring fairness in the utilization of government-funded public facilities, such as co-working spaces, sports fields, and meeting rooms, is imperative to accommodate all citizens. However, meeting these requirements poses a significant challenge due to the high costs associated with maintaining digital infrastructure, employee wages, and cybersecurity expenses. Fortunately, Blockchain smart contracts present an economical and secure solution for managing digital infrastructure. They offer a pay-per-transaction schema, immutable transaction records, and role-based data updates. Despite these advantages, public blockchains raise concerns about data privacy since records are publicly readable. To address this issue, this study proposes a privacy-preserving mechanism for public facilities' reservation systems. The approach involves encrypting the reservation table with fully-homomorphic encryption (FHE). By employing FHE with binary masking and polynomial evaluation, the reservation table can be updated without decrypting the data. Consequently, citizens can discreetly book facilities without revealing their identities and eliminating the risk of overlapping schedules. The proposed system allows anyone to verify reservations without disclosing requested data and table contents. Moreover, the system operates autonomously without the need for human administration, ensuring enhanced user privacy.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



## Corresponding Author:

Akbari Indra Basuki

Research Center for Artificial Intelligence and Cyber Security, National Research and Innovation Agency

Cisitu St. No. 21, Bandung, West Java, Indonesia

Email: akbari.indra.basuki@brin.go.id

## 1. INTRODUCTION

Public facilities funded by taxes should prioritize fairness and transparency in their management. In many developing countries, upholding these values faces obstacles due to prevalent corrupt behavior and the substantial cost associated with maintaining information technology (IT) infrastructure. A conventional IT-based solution typically involves three key cost centers: the IT infrastructure itself, employee wages, and cybersecurity expenses. This significant financial burden has impeded the widespread adoption of IT-based solutions in these regions.

With its rapid growth and strong community support, blockchain technology now provides cost-effective and secure solutions for managing public facilities. Various techniques, such as proof-of-stake [1], [2], sharing [3], and second-layer roll-up [4], [5], have significantly reduced the pay-per-transaction cost, making it hundreds or even thousands of times cheaper than the initial generation of blockchain (proof-of-work). In terms of security, blockchain outperforms typical server/cloud-based technology. Key advantages include distributed data replication, immutable transaction records, transparent ledger, and non-reputability

[6]–[8]. Consequently, a blockchain system does not necessitate additional security measures. Its distributed data storage makes it impervious to distributed denial-of-service (DDoS) attacks, ensuring a system availability of 100%. Attempting to bring down all participating nodes is nearly impossible. Moreover, altering transaction records is highly unlikely, as an attacker would need to compromise at least 50%+1 of the blockchain nodes. This robust security framework establishes blockchain as a reliable and resilient solution for managing public facilities.

The primary concern associated with public blockchains revolves around safeguarding data privacy. Numerous approaches have been suggested to address this issue, including the integration of zero-knowledge proof (ZKP) [9]–[12], multi-party computation [13]–[15], secret sharing [16], or fully homomorphic encryption [10], [11], [17]–[19]. In study [9], a zero-knowledge proof is employed for a ride-hailing application utilizing zoning for pick-up and drop-off areas, showcasing a practical application of ZKP. Additionally, Wang *et al.* [13] introduces multi-party computation to enhance user privacy in energy storage distribution. Concise overviews of blockchain-utilizing multi-party computation can be found in [14] and [15]. Despite the privacy-preserving capabilities of these methods, they exhibit less flexibility in preserving user privacy compared to fully homomorphic encryption (FHE). FHE facilitates addition and multiplication over encrypted data, making it versatile across various research fields. Its application extends to blockchain-related scenarios such as housing rental [10], parking recommendations [11], and encrypted machine learning models [18], [19]. Furthermore, FHE plays a crucial role in securing data privacy in image watermarking computations and enabling transparent image licensing over blockchain networks [20], [21].

This study presents a reservation system structured around a reservation table schema strengthened with leveled fully homomorphic encryption [22] for ensuring data privacy. The preference for the reservation table model stems from two key reasons. Firstly, it is user-friendly and easily manageable by individuals. Secondly, it facilitates encrypted computation using FHE for reservation updates and conflict testing. The utilization of FHE is advantageous as it enables public observation of the reservation process alongside its privacy protection benefits. The proposed approach involves employing a binary masking technique and polynomial evaluation to prevent conflicting updates on the encrypted reservation system. Given that smart contracts lack support for FHE, FHE computation is run by the facility's IoT devices without requiring human intervention.

The objectives of the proposed methods encompass two primary goals. Firstly, it allows users to book and verify their requests without divulging their identity or preferred facility and reservation time. Secondly, it facilitates public verifiability of the reservation process by referencing encrypted data stored in the public ledger. This study pioneers the concept of a confidential and auditable reservation table schema using a public blockchain and FHE. Previous approaches have primarily relied on consortium blockchains to uphold user privacy [23]–[34]. While effective in preserving privacy, these methods often lack transparency in the reservation process. In contrast, public blockchain methods either fail to adequately protect user privacy [35]–[38] or neglect to implement a verifiable reservation table schema [10], [11], [39]–[41]. The key contributions of our work are outlined below: i) Proposed a confidential and verifiable reservation table schema utilizing smart contracts on public blockchains, ii) Introduced a method for checking reservation conflicts that operates on encrypted reservation tables without necessitating data decryption, and iii) Developed an autonomous reservation model devoid of human verification, employing a blockchain smart contract and an IoT device to mitigate insider attacks that could compromise user data.

The paper is structured by first examining the limitations of FHE computation and integrating its implications into the design of smart contracts and the preservation model in section 2. Following that, the effectiveness of the binary masking method in resolving overlapping reservations is assessed, and the implementation results are discussed in section 3. In conclusion, section 4 summarizes the key findings of the study.

## 2. PROPOSED METHOD

In this study, the discussion on privacy preservation is limited to the protection of users' identity and reservation preferences. User identity encompasses citizenship information containing sensitive personal data, while reservation preferences include requested facilities and reserved times. These preference data are connected to users' daily routines, such as work schedules, sports activities, and other engagements. Any leakage of this information could pose a threat to user safety, potentially enabling malicious parties to stalk or harm users at specific moments. This section elaborates on the use of FHE to protect the aforementioned data, particularly when transactions are recorded in public blockchains. Initially, the constraints associated with FHE computation are outlined. Next, the section discusses its implications on smart contract design and the preservation model aimed at securing user identity and preferences.

## 2.1. FHE limitations and countermeasures

The proposed method employs FHE to safeguard user data privacy by encrypting the data within blockchain transactions. Accordingly, the protection of data privacy needs to be structured considering the limitations of FHE. Two notable drawbacks of FHE include its restricted computational capacity and the substantial size of encrypted data.

### 2.1.1. Limited computation ability

Fully homomorphic encryption (FHE) facilitates computation over encrypted data, providing results similar to those obtained through computation on plain data. If  $E_A$  and  $E_B$  represent the encrypted values of  $A$  and  $B$  (1), (2), the properties of fully homomorphic encryption can be defined as follows.

- The decryption results of the addition  $E_A$  and  $E_B$  with key  $s$  is approximately equal to the addition of  $A$  and  $B$  since the computation noise of encrypted addition is relatively small (3).

$$E_A = FHE_{Encrypt}(A) = C_A = C_{A0}, C_{A1} \quad (1)$$

$$E_B = FHE_{Encrypt}(B) = C_B = C_{B0}, C_{B1} \quad (2)$$

$$\begin{aligned} FHE_{Decrypt}(E_A + E_B) &= C_{add,0} + C_{add,1} \cdot s \\ &= (C_{A0} + C_{B0}) + (C_{A1} + C_{B1}) \cdot s \\ &= (C_{A0} + C_{A1} \cdot s) + (C_{B0} + C_{B1} \cdot s) \\ &= FHE_{Decrypt}(E_A \cdot s) + FHE_{Decrypt}(E_B \cdot s) \approx A + B \end{aligned} \quad (3)$$

- The decryption results of multiplication  $E_A$  and  $E_B$  is approximately equal to the multiplication of  $A$  and  $B$  if the computation noise remains below the specified limit budget. The decryption of two encrypted multiplication results can be regarded as the evaluation of the secret key  $s$  on a two-degree polynomial (4).

$$\begin{aligned} FHE_{Decrypt}(E_A \times E_B) &= (C_{A0} + C_{A1} \cdot s) \times (C_{B0} + C_{B1} \cdot s) \\ &= C_{A0} C_{B0} + C_{A0} C_{B1} s + C_{B0} C_{A1} s + C_{A1} C_{B1} s^2 \\ &= C_{A0} C_{B0} + (C_{A0} C_{B1} + C_{B0} C_{A1}) s + C_{A1} C_{B1} s^2 \\ &= d_0 + d_1 s + d_2 s^2 \approx A \times B \end{aligned} \quad (4)$$

- The outcomes of encrypted multiplication will exhibit exponential growth with each successive multiplication. Consequently, once the computation noise ( $e$ ) surpasses the predetermined budget limit, decrypting the encrypted data accurately becomes unattainable (5).

$$Decryption = \begin{cases} e < \text{limit budget}, FHE_{Decrypt}(E_A \times E_B) \cong A \times B \\ e \geq \text{limit budget}, FHE_{Decrypt}(E_A \times E_B) \neq A \times B \end{cases} \quad (5)$$

Leveled FHE [22] imposes restrictions on the consecutive computations applicable to encrypted data. With each calculation, the noise level tends to increase. While raising the polynomial modulus of FHE could expand the limit budget, this option is less favorable due to its associated increase in computation time and memory space. To tackle this constraint, the chosen approach involves setting the reservation timetable on a daily basis. This strategy limits consecutive computations on the encrypted data to only include the data encrypted for the corresponding date.

### 2.1.2. The huge size of encrypted data

Given the limited transaction size of blockchain, storing encrypted data directly into the blockchain ledger becomes impractical due to the high cost of gas and the constraint of the block limit. As an alternative, the encrypted reservation data is stored in the interplanetary file system (IPFS) [42]. IPFS, a distributed protocol, ensures data integrity using the distributed hash table (DHT). By pinning the IPFS hash in cloud storage, anyone can retrieve the data using the corresponding hash value. The 46-digit length of the IPFS hash is suitable for blockchain transactions, costing only 46 bytes of data writing.

## 2.2. Smart contract design

The Ethereum-based smart contract lacks native support for fully-homomorphic encryption computation. Nonetheless, ongoing efforts are being made to integrate FHE into the smart contract [43]. In this study, FHE computation is executed on the end-user and facility's IoT devices. The end-user device may be a smartphone or a desktop computer. Simultaneously, the facility device can take the form of an IoT

device or a small form factor PC connected to the blockchain ledger through a third-party gateway service (such as Infura, QuickNodes, and GetBlocks).

Figure 1 depicts the sequence diagram of the system. The smart contract records bidirectional transactions to ensure transparency. These transactions encompass two types: request transactions, sent by the user, and reply to transactions, sent by the end device of the facility.

The structure of the smart contract is presented in Table 1. Each smart contract corresponds to a single facility. Users can initiate a reservation by sending a transaction to the smart contract. The contract retains static information crucial for reservation purposes, such as  $F_{prm}$ , denoting the facility identity number, and the corresponding reservation date. Additionally,  $F^{PK}$  stores the facility's public key for FHE computation. Meanwhile, the reservation table ( $R_{TT}$ ) serves as the dynamic variable, storing daily reservation records with FHE encryption. The  $Rsv$  stores the reservation data and its requested, approved, or rejected status.

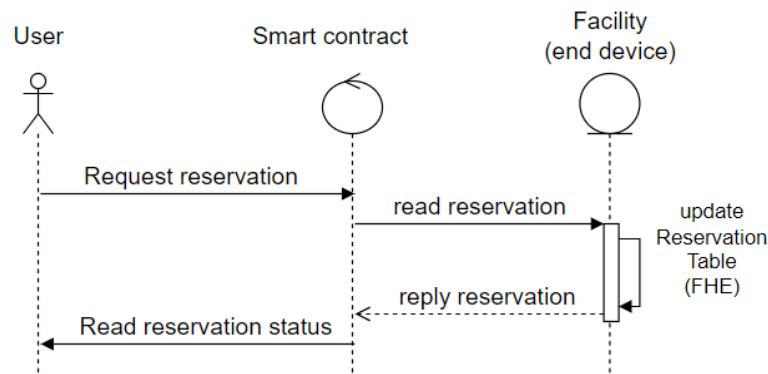


Figure 1. Sequence diagram of reservation model using smart contract of public blockchain

Table 1. Smart contract structure: data states, structures, and functions

Name	Type	Description
$F_{prm}$	uint256	Facility parameter
$F^{PK}$	string	Facility's public key
$R_{TT}$	map(int: String)	Reservation table (indexed with Facility ID and the date: FIDYYYYMMDD)
$Rsv$	struct	Storing reservation data, indexed by a counter
$reqCtr$	map(int: Rsv)	Mapped array to index reservation request
$request\_reservation$	function	Send reservation request (by user)
$reply\_reservation$	function	Send reservation reply (by device)

### 2.3. User identity preservation

Our proposed method aims to decouple user identity from the blockchain address to enhance privacy protection. Although the address does not directly link to the user identity, potential vulnerabilities exist through side-channel analysis. A side-channel attack could leverage transfer history to infer the ownership or wealth of a particular account. To counter this, our method employs FHE-based identity protection, termed virtual identity ( $V_{ID}$ ). Instead of referencing a specific address, our approach relies on  $V_{ID}$  as the user identifier. The user can send a transaction using any shared address and insert the  $V_{ID}$  into the transaction parameters.

The  $V_{ID}$  is an encrypted array comprising a citizen ID ( $C_{ID}$ ) and a random value ( $Rnd$ ) (5). The virtual identity  $V_{ID}$  is secure as it undergoes encryption using the end device's public key. Consequently, only the end device possesses knowledge of the actual identity of the sender  $C_{ID}$  (8).

A standard transaction initiated by the user comprises a tuple of four data: virtual identity ( $V_{ID}$ ), facility parameter ( $F_{prm}$ ), reservation request ( $R_R$ ), and computation proof ( $CP$ ) (9). The  $F_{prm}$  is not encrypted since it indexes the timetable data ( $R_{TT}$ ) in the smart contract (6). This schema ensures the non-disclosure of user identity, as the virtual identifier  $V_{ID}$  was employed instead of the blockchain address. Since  $V_{ID}$  is a one-time-use ID, inter-correlation analysis is rendered ineffective for tracing user information. In contrast, the reservation request ( $R_R$ ) contains fine-grain information regarding the reservation timetable (hours or minutes); thus, encryption is mandatory (7).

The computation proof ( $CP$ ) stores the anticipated computation result for updating the reservations table and is computed by the requester. Since it contains the expected result as computation proof, it must undergo encryption and remain verifiable. In the absence of a reservation conflict, the new reservation table must adopt this proof ( $CP$ ) as its new reservation table value ( $R_{TT}$ ). Consequently, this approach prevents a compromised facility device from maliciously or incorrectly updating the reservation table without referencing the reservation request. The proposed method restricts the authority of the facility device solely to checking for reservation conflicts and approving/rejecting the reservation table, without being involved in computing the reservation table update. The detailed mechanism for securing reservation requests and generating computation proof is outlined in subsection 2.3.

The transaction reply encompasses four parameters: a  $request\_ID$  denoting the reservation index,  $F_{prm}$  indicating the timetable and sub-facility, an approval variable signifying the conflict status, and  $U_{TT}$  representing the reply to data. In the event of a reservation conflict, the approval parameter is set to false. Consequently, the contract refrains from updating the reservation table. Conversely, if no conflict arises, the approval value is true, prompting the contract to update the reservation timetable value ( $R_{TT}$ ) by overwriting it with the stored computation proof ( $CP$ ) provided by the requester. Ultimately, irrespective of the conflict value, the user receives an encrypted copy of the reservation table ( $R_{TT}$ ) value using the user's public key ( $U^{PK}$ ) as  $U_{TT}$ .

$$V_{ID} = FHE_{Encrypt}([C_{ID}, Rnd]) \quad (5)$$

$$F_{prm} = Concatenate(F_{ID} + date) \quad (6)$$

$$R_R = FHE_{Encrypt}([reservation\_request]) \quad (7)$$

$$C_{ID} = FHE_{Decrypt}(IPFS(V_{ID})) \quad (8)$$

$$T_{req} = [IPFS(V_{ID}), F_{prm}, IPFS(R_R), IPFS(CP)] \quad (9)$$

$$T_{rpl} = [Request\_ID, F_{prm}, approval, IPFS(U_{TT})] \quad (10)$$

The proposed method encounters a limitation wherein citizens must have registered and recorded in the system beforehand. It is essential to note that, for the purposes of this study, user registrations were excluded as a limitation. However, it is worth mentioning that the registration process can be easily implemented through two-factor authentication (TFA) via email or instant messaging.

## 2.4. User preferences preservation

User preferences entail the specific time slot and room/field of a facility the user aims to book. It is imperative that the system safeguards this information in a secretive yet traceable manner, where only the user possesses the information. Simultaneously, anyone should be able to verify that the reservation process has been executed accurately. Preserving user preferences involves both data preservation and process preservation.

### 2.4.1. Data preservation

Utilizing a facility reservation table, the computation is executed using a serialized vector format for efficiency in Figure 2. Recognizing the limited noise accumulation of level FHE, the risk of noise overflow was mitigated by restricting the number of allowed computations. By partitioning the reservation table on a daily basis, data updates are exclusively applied to the corresponding date rather than aggregated time series data. Consequently, the frequency of repetitive computations over a specific encrypted vector decrease, thereby reducing noise growth.

Data preservation serves two primary objectives: informing the user of the conflicting status and updating the reservation table stored in the facility's smart contract. For conflict notification, the vector data ( $V_R$ ) is encrypted using the user's public key ( $U^{PK}$ ) as  $E^U V_R$  (11). To facilitate updating, the vector data ( $V_R$ ) must be encrypted using the facility's public key ( $F^{PK}$ ) as  $E^F V_R$  (12). Consequently, the data remains secure, as it can only be decrypted by the respective public key owner.

$$E^U V_R = FHE_{Encrypt}(V_R, U^{PK}) \quad (11)$$

$$E^F V_R = FHE_{Encrypt}(V_R, F^{PK}) \quad (12)$$

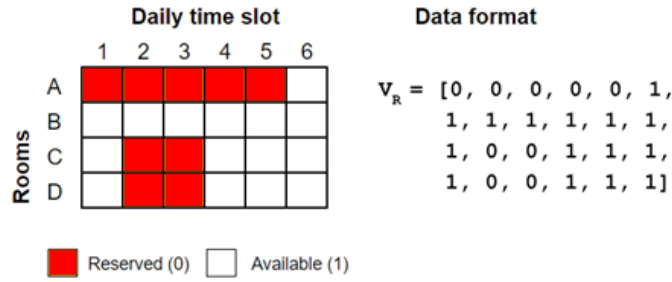


Figure 2. An example of reservation table representation in vector data format ( $V_R$ )

**2.4.2. Process preservation**

While FHE offers an advantage in secret computation verifiability, it is constrained to addition and multiplication operations. However, the reservation process necessitates conditional evaluation. To address this, polynomial evaluation was employed as a form of binary masking. This binary masking normalizes computation results into a binary format, representing the facility's occupancy status (value=0) or availability status (value=1). Careful selection of a suitable polynomial equation is crucial to aligning with the reservation procedure.

The polynomial equation for the reservation update can be deduced from the computation table of the reservation table in Table 2. No alterations are made to the reservation table if no reservation has been placed. In cases where a reservation is made for a previously reserved timeslot, the reservation is disregarded, and no modifications are implemented. If the intended schedule is available (1), the reservation must update the reservation table and alter its status to occupied (0). The polynomial function ( $f_U$ ) that complies to this rule is  $y = 0.5x^2 + 0.5x$  as shown in Figure 3.

Detecting conflicting schedules can be accomplished using a distinct polynomial equation. The normalization table, illustrated in Table 2, facilitates the identification of conflicting schedules without the need for decrypting the data. The objective is to designate conflicting occurrences as 1 and non-conflicting as 0. The suitable polynomial for this normalization is  $f_C(x) = 0.5x^2 - 0.5x$  as shown in Figure 3.

**Table 2. Computation table for updating the reservation table and checking the conflicting schedule**

Availability	Reservation (yes/no)	Reservation result (x = A - R)	Polynomial evaluation, $f_U(x)$ (Updated availability)	Polynomial evaluation, $f_C(x)$ (Conflicting result)
1	0	1	1	0
1	1	0	0	0
0	0	0	0	0
0	1	-1	0	1

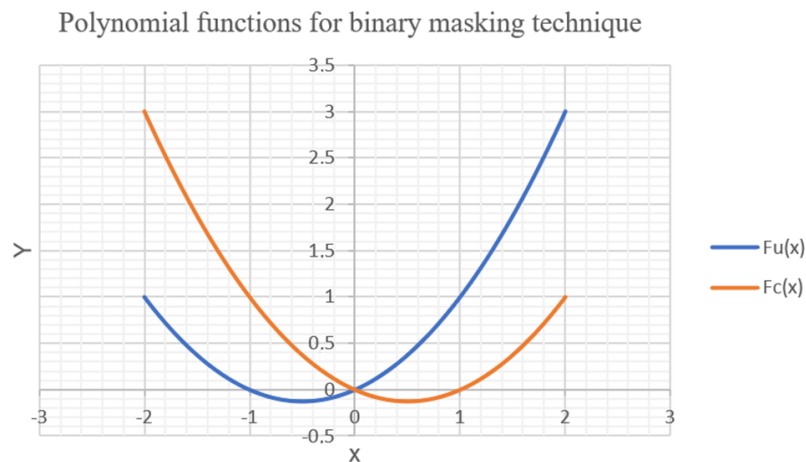


Figure 3. Polynomial functions for binary masking for reservation update ( $f_U$ ) and conflict checking ( $f_C$ ),  $f_U(x) = 0.5x^2 + 0.5x$  and  $f_C(x) = 0.5x^2 - 0.5x$

Expressed in mathematical notation, the reservation process can be represented as vector subtraction. The computation initiates by subtracting the encrypted reservation vector sent by the user ( $E^F V_U$ ) from the encrypted reservation timetable vector of the reserved facility ( $E^F R_{TT}$ ) (13). The proof of the reservation table update is then documented in the smart contract as the proof of computation ( $E^F CP$ ) (9).

The computation yields negative numbers if the intended time slot has been reserved in Table 2. In such instances, the updating polynomial ( $f_U$ ) will standardize the subtraction result into a binary format using (14)-(16). Subsequently, the facility device assesses conflicting schedules through the conflicting polynomial evaluator ( $f_C$ ) (17) in Table 2. If the end device detects no conflicting result ( $C_{Value} = 0$ ) (18), the new value ( $E^F CP$ ) will be considered as the new reservation table ( $E^F R_{TT}$ ); otherwise, the table remains unaltered (19). Finally, the user is notified of the encrypted final reservation table using their public key ( $U^{PK}$ ) (20).

$$E^F V_{Rsv} = E^F R_{TT} - E^F V_U \quad (13)$$

$$f_U(x) = 0.5x^2 + 0.5x \quad (14)$$

$$E^F CP = f_U(E^F V_{Rsv}) \quad (15)$$

$$E^F CP = 0.5 (E^F V_{Rsv})^2 + 0.5 (E^F V_{Rsv}) \quad (16)$$

$$E^F C_{Value} = 0.5 (E^F V_{Rsv})^2 - 0.5 (E^F V_{Rsv}) \quad (17)$$

$$C_{Value} = \sum_{i=0}^N (FHE_{Decrypt}(E^F C_{Value})) \quad (18)$$

$$E^F R_{TT} = \begin{cases} E^F R_{TT} & | C_{Value} > 0 \\ E^F CP & | C_{Value} = 0 \end{cases} \quad (19)$$

$$E^U R_{TT} = FHE_{Encrypt}(R_{TT}, U^{PK}) \quad (20)$$

### 3. RESULTS AND DISCUSSION

#### 3.1. Implementation results

The privacy preservation method was implemented using the binary masking technique to validate reservation table updates. Equations (14)-(16) are utilized to confirm the correctness of the computation for a reservations table update. The implementation codes are available in [44].

Table 3 displays the test results of reservation updates and conflicting reservations. The initial case involves consecutive reservations for a specific facility room (room A). The first reservation (transaction-1) secures the first and second timeslots, while the subsequent reservation (transaction-2) attempts to book the third and fourth timeslots. As both reservations present no conflict, the reservation table is successfully updated with the respective proof stored in IPFS networks. This proof includes an encrypted reservation table that can be verified by anyone through computation (13)-(16). In the subsequent reservation (transaction-3), another user endeavors to book time slots 1, 3, and 5, overlapping with previous requests. In this case, the reservation table remains unaltered, and the current one is encrypted and transmitted to the user via IPFS networks for their consideration.

The second case involves a reservation request for the next day. Given that the reservation table is stored daily, the reservation (transaction-4) does not conflict with previous transactions. Both reservations for room A and room B successfully updated the reservations table. However, the final reservation transaction (transaction-5) overlaps with the preceding one (transaction-4). Consequently, instead of updating the new reservation table, it dispatches the encrypted reservation table to the user via IPFS networks. All IPFS hashes in Table 3 are accessible by accessing IPFS networks through a web browser with the URL format: [https://ipfs.io/ipfs/{hash\\_value}](https://ipfs.io/ipfs/{hash_value}). For instance, the final reservation table of day 1 can be retrieved using <https://ipfs.io/ipfs/QmRTHWLQREDimhU4jGyac2c86uPrKXWHyr85VRxQa3DjZF>.

The assessment of blockchain transaction costs is conducted using Polygon networks. The evaluation involves implementing the system as illustrated in Figure 1, with transaction parameters specified in (9), (10). The smart contract transaction can be accessed at study [44] and is deployed at `0x267EC91106e1C89886279EBc4140b4d59ec43B09` on the Polygon mainnet. The results underscore the economic benefits of blockchain smart contracts. The one-time deployment fee amounts to approximately 2500 IDR. Meanwhile, regular transactions, including request, reply, and add user, incur costs of less than 400 IDR.

Table 3. Testing table for FHE-encrypted reservation model using binary format and binary masking

Reservation	Data	Room A	Room B	Room C	Room D	Status
Transaction-1, Room A, Day 1	Prev. table	111111	111111	111111	111111	
	Booking	110000	000000	000000	000000	
	Post. Table	001111	111111	111111	111111	Updated
Transaction-2, Room A, Day 1	CP (IPFS)	QmPLNK4D4hnXyyJKnAouYSfRzNXVKu7Fd6JVfXjFpqpun8				New table
	Prev. table	001111	111111	111111	111111	
	Booking	001100	000000	000000	000000	
Transaction-3, Room A, Day 1	Post. Table	000011	111111	111111	111111	Updated
	CP (IPFS)	QmRTHWLQREDimhU4jGyac2c86uPrKXWHyr85VRxQa3DjZF				New table
	Prev. table	000011	111111	111111	111111	
Transaction-4, Room A & B, Day 2	Booking	101010	000000	000000	000000	
	Post. Table	000011	111111	111111	111111	Conflict
	U <sub>TT</sub> (IPFS)	QmQTRAY6t9gcifsT4xZ8owQssqQ8UcAEuWRbB4A88ubh9D				Inform user
Transaction-5, Room B, Day 2	Prev. table	111111	111111	111111	111111	
	Booking	111000	111000	000000	000000	
	Post. Table	000111	000111	111111	111111	Updated
Transaction-5, Room B, Day 2	CP (IPFS)	QmbneA11oUGz1KeXzbXwwVuZQegRgejRL6W33C16Rt2YsH				New table
	Prev. table	000111	000111	111111	111111	
	Booking	000000	111111	000000	000000	
Transaction-5, Room B, Day 2	Post. Table	000111	000111	111111	111111	Conflict
	U <sub>TT</sub> (IPFS)	QmT3XH8ePYyN6XjxP5itxDWgWd46ZvN7eA8qCkQvXiPS5				Inform user

### 3.2. Security evaluation

Employing an encrypted reservation table for public facility reservations guarantees public monitoring of the reservation process. The primary threats to the online reservation system include reservation fairness, sniping attacks, and occupancy transparency. These issues can be effectively addressed by integrating an FHE-encrypted reservation table with blockchain technology.

Securing encrypted reservations through blockchain and IPFS ensures system fairness. In contrast to a centralized reservation system where administrators have comprehensive control and information access, blockchain networks rely on HEX-format addresses, preserving user privacy. The reservation system is not under the control of a specific party but rather an automated contract deployed on a blockchain network. The likelihood of discreetly rejecting a specific group or ethnicity from reserving the facility is nearly eliminated. As a result, anyone can submit reservation requests to the deployed contract on a fair basis.

However, the use of public blockchain introduces a new vulnerability known as sniping. A sniping attack aims to obstruct a specific party from securing their desired facility or time slots. In the Blockchain context, if two transactions vie for the same facility and time slot, the miner will execute the first transaction they pick up. Since all transactions are stored in the mempool before a new block creation and are accessible to anyone, it becomes possible for a particular group to consistently outbid others' requests by submitting transaction requests with higher gas fees. Given that miners typically prioritize transactions with higher gas fees, they can thwart other requests by consistently submitting similar ones with higher gas fees. The conventional solution to this problem involves integrating the payment system into reservation requests. However, this approach proves ineffective in real-world scenarios where economic imbalances persist. In such cases, a wealthier party can persistently disrupt the reservation process for less affluent individuals by submitting duplicate transactions with higher gas fees.

Our proposed method employs FHE encryption on the reservation request, significantly reducing the susceptibility to sniping attacks. The reservation request data is encrypted using the facility's public key, ensuring that only the facility's end-device can decrypt and identify the reserved facility. As the reply transaction corresponds to the request transaction, they cannot be recorded in the same block. Consequently, adversaries have minimal opportunity to execute sniping attacks. The only potential way for the attack to succeed is to blindly guess the reservation, a task that proves challenging.

The proposed FHE-encrypted reservation table ensures reservation transparency, as anyone can trace and verify the computation update of the encrypted reservation table. The requester receives information about the latest reservation table, regardless of the reservation status. In the event of a conflicting reservation, the user and anyone else can recompute the reservation update to determine if it produces the same result (13)-(16). If the computation does not match, it suggests a compromise in the facility's end-device, triggering an investigation request for device auditing. By recording the encrypted reservation table on public blockchains, anyone can confirm the correctness of the computation while the requester can verify the occupancy status of public facilities.

### 3.3. Contribution and future directions

Most existing works in blockchain-based reservation systems have drawbacks in data privacy or reservation process transparency as shown in Table 4. Solutions using consortium blockchain excel in



preserving user privacy but lack public observability since they utilize permission-based ledgers [23]–[34]. Conversely, works employing public blockchain offer public monitoring features at the expense of user privacy [35]–[38]. According to the literature study, five works satisfy both data privacy protection and public observability [10], [11], [39]–[41]. However, only three of them support verifiable reservation or rental updates [10]–[12]. Nevertheless, these works do not incorporate a reservation table schema. The study in [39] is the only one that employs a table-like reservation schema called an exchange pool, but it does not feature provable reservation updates. Our study is the first to guarantee four aspects simultaneously: a reservation table schema, privacy protection, public observability, and provable reservation updates.

Table 4. Comparison of Blockchain-based reservation system

Methods	Year	Reservation mechanism			Privacy protection		Transparency	
		Agreement Only	Secure transaction (ST)	Reservation table	User	Reservation data	Public monitoring	Provable reservation
[28], [32]	2019		✓		✓	✓		
[24]–[27]	2020		✓		✓	✓		
[38]	2020	✓					✓	
[39]	2020		✓	✓	✓	✓	✓	
[10], [11]	2021		✓		✓	✓	✓	✓
[35], [37]	2021	✓					✓	
[36]	2021	✓	✓				✓	
[40]	2021		✓		✓		✓	
[23], [29], [30]	2022		✓		✓	✓		
[34]	2022	✓					✓	
[41]	2022		✓		✓	✓	✓	✓
[31]	2023		✓		✓	✓		
[33]	2024		✓		✓	✓		
Proposed	2024		✓	✓	✓	✓	✓	✓

The four aspects are crucial for maintaining the reservation system's integrity. They safeguard user privacy while enabling public monitoring of the reservation process. Moreover, they empower users to verify the system's integrity through encrypted computation and manage reservations using a reservation table. The proposed method will have a significant impact on various reservation systems that prioritize system traceability and accountability, such as smart parking, housing, car rental, and insurance.

The study has a limitation due to the lack of direct integration of FHE computation with the smart contract. Encouraging further development to integrate FHE as native smart contract features, as demonstrated in [43], would ease the adoption of FHE computation. Considering the use of ZKP [10]–[12] could be a more effective alternative for preserving user identity.

#### 4. CONCLUSION

This study involves the implementation and evaluation of a secret and verifiable reservation table schema for reserving public facilities via Blockchain networks. It ensures users can generate non-conflicting reservations while allowing them to verify the correctness of the reservation process. The reservation approval is automated using an IoT device, eliminating the potential for insider attacks by operators. Through the use of the FHE-encrypted reservation table schema, our proposed method simultaneously ensures privacy protection, public observability, and provable reservation updates. Additionally, it incurs minimal transaction costs, which are negligible compared to the actual reservation fee. The proposed reservation system holds high potential for diverse applications such as smart parking, housing, car rental, and insurance.

#### REFERENCES





- [1] P. Gazi, A. Kiayias, and D. Zindros, "Proof-of-stake sidechains," in *2019 IEEE Symposium on Security and Privacy (SP)*, May 2019, pp. 139–156, doi: 10.1109/SP.2019.00040.
- [2] B. Sriman, S. Ganesh Kumar, and P. Shamili, "Blockchain technology: consensus protocol proof of work and proof of stake," in *Advances in Intelligent Systems and Computing*, Springer Singapore, 2020, pp. 395–406.
- [3] D. R. Lee, Y. Jang, and H. Kim, "Poster: a proof-of-stake (PoS) Blockchain protocol using fair and dynamic sharding management," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, Nov. 2019, pp. 2553–2555, doi: 10.1145/3319535.3363254.

- [4] T. Lavour, J. Lacan, and C. P. C. Chanel, "Enabling Blockchain services for IoE with Zk-rollups," *Sensors*, vol. 22, no. 17, Aug. 2022, doi: 10.3390/s22176493.
- [5] J. Kanani, S. Nailwal, and A. Arjun, "Matic whitepaper," *Polygon*, Bengaluru, India, 2021.
- [6] S. Alam *et al.*, "An overview of Blockchain and IoT integration for secure and reliable health records monitoring," *Sustainability*, vol. 15, no. 7, Mar. 2023, doi: 10.3390/su15075660.
- [7] J. Golosova and A. Romanovs, "The advantages and disadvantages of the Blockchain technology," in *2018 IEEE 6th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*, Nov. 2018, pp. 1–6, doi: 10.1109/AIEEE.2018.8592253.
- [8] A. Farouk, A. Alahmadi, S. Ghose, and A. Mashatan, "Blockchain platform for industrial healthcare: Vision and future opportunities," *Computer Communications*, vol. 154, pp. 223–235, Mar. 2020, doi: 10.1016/j.comcom.2020.02.058.
- [9] M. Baza, N. Lasla, M. M. E. A. Mahmoud, G. Srivastava, and M. Abdallah, "B-ride: ride sharing with privacy-preservation, trust and fair payment atop public Blockchain," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1214–1229, Apr. 2021, doi: 10.1109/tNSE.2019.2959230.
- [10] Z. Li, M. Alazab, S. Garg, and M. S. Hossain, "PriParkRec: privacy-preserving decentralized parking recommendation service," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 5, pp. 4037–4050, May 2021, doi: 10.1109/tvt.2021.3074820.
- [11] R. Yu, Z. Wang, C. Zhang, and S. Guan, "A secure Blockchain-based housing rental platform," in *2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, Jun. 2021, pp. 2049–2053, doi: 10.1109/IMCEC51613.2021.9482058.
- [12] M. Baza, R. Amer, A. Rasheed, G. Srivastava, M. Mahmoud, and W. Alasmaly, "A Blockchain-based energy trading scheme for electric vehicles," in *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, Jan. 2021, pp. 1–7, doi: 10.1109/CCNC49032.2021.9369517.
- [13] N. Wang, S. C.-K. Chau, and Y. Zhou, "Privacy-preserving energy storage sharing with Blockchain and secure multi-party computation," *ACM SIGEnergy Energy Informatics Review*, vol. 1, no. 1, pp. 32–50, Nov. 2021, doi: 10.1145/3508467.3508471.
- [14] S. Wu, J. Li, F. Duan, Y. Lu, X. Zhang, and J. Gan, "The survey on the development of secure multi-party computing in the Blockchain," in *2021 IEEE Sixth International Conference on Data Science in Cyberspace (DSC)*, Oct. 2021, pp. 1–7, doi: 10.1109/DSC53577.2021.00008.
- [15] H. Zhong, Y. Sang, Y. Zhang, and Z. Xi, "Secure multi-party computation on Blockchain: an overview," in *Communications in Computer and Information Science*, Springer Singapore, 2020, pp. 452–460.
- [16] J. Cha, S. K. Singh, T. W. Kim, and J. H. Park, "Blockchain-empowered cloud architecture based on secret sharing for smart city," *Journal of Information Security and Applications*, vol. 57, Mar. 2021, doi: 10.1016/j.jisa.2020.102686.
- [17] L. Zhou, L. Wang, T. Ai, and Y. Sun, "BeeKeeper 2.0: confidential Blockchain-enabled IoT system with fully homomorphic computation," *Sensors*, vol. 18, no. 11, Nov. 2018, doi: 10.3390/s18113785.
- [18] J. Chen, K. Li, and P. S. Yu, "Privacy-preserving deep learning model for decentralized VANETs using fully homomorphic encryption and Blockchain," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11633–11642, Aug. 2022, doi: 10.1109/tits.2021.3105682.
- [19] S. Yaji, K. Bangera, and B. Neelima, "Privacy preserving in Blockchain based on partial homomorphic encryption system for AI applications," in *2018 IEEE 25th International Conference on High Performance Computing Workshops (HiPCW)*, Dec. 2018, pp. 81–85, doi: 10.1109/HiPCW.2018.8634280.
- [20] A. Basuki, I. Setiawan, and D. Rosiyadi, "Preserving privacy for Blockchain-driven image watermarking using fully homomorphic encryption," in *Proceedings of the 2021 International Conference on Computer, Control, Informatics and Its Applications*, Oct. 2021, pp. 151–155, doi: 10.1145/3489088.3489130.
- [21] D. Rosiyadi, A. I. Basuki, T. I. Ramdhani, H. Susanto, and Y. H. Siregar, "Approximation-based homomorphic encryption for secure and efficient blockchain-driven watermarking service," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 4, pp. 4388–4400, Aug. 2023, doi: 10.11591/ijece.v13i4.pp4388-4400.
- [22] H. Chen, K. Laine, and R. Player, "Simple encrypted arithmetic library - SEAL v2.1," in *Lecture Notes in Computer Science*, Springer International Publishing, 2017, pp. 3–18.
- [23] J. Wang *et al.*, "BPR: Blockchain-enabled efficient and secure parking reservation framework with block size dynamic adjustment method," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 3, pp. 3555–3570, Mar. 2023, doi: 10.1109/TITS.2022.3222960.
- [24] M. M. Badr, W. Al Amiri, M. M. Fouda, M. M. E. A. Mahmoud, A. J. Aljohani, and W. Alasmaly, "Smart parking system with privacy preservation and reputation management using Blockchain," *IEEE Access*, vol. 8, pp. 150823–150843, 2020, doi: 10.1109/access.2020.3016945.
- [25] C. Zhang *et al.*, "BSFP: Blockchain-enabled smart parking with fairness, reliability and privacy protection," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 6578–6591, Jun. 2020, doi: 10.1109/tvt.2020.2984621.
- [26] S. Cao, S. Dang, X. Du, M. Guizani, X. Zhang, and X. Huang, "An electric vehicle charging reservation approach based on Blockchain," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, Dec. 2020, pp. 1–6, doi: 10.1109/GLOBECOM42002.2020.9322093.
- [27] W. Al Amiri, M. Baza, K. Banawan, M. Mahmoud, W. Alasmaly, and K. Akkaya, "Towards secure smart parking system using Blockchain technology," in *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, Jan. 2020, pp. 1–2, doi: 10.1109/CCNC46108.2020.9045674.
- [28] W. Al Amiri, M. Baza, K. Banawan, M. Mahmoud, W. Alasmaly, and K. Akkaya, "Privacy-preserving smart parking system using Blockchain and private information retrieval," in *2019 International Conference on Smart Applications, Communications and Networking (SmartNets)*, Dec. 2019, pp. 1–6, doi: 10.1109/SmartNets48225.2019.9069783.
- [29] C. Wang, W. Jia, and Y. Chen, "Housing rental scheme based on redactable Blockchain," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–10, Mar. 2022, doi: 10.1155/2022/1137130.
- [30] S. Auer, S. Nagler, S. Mazumdar, and R. R. Mukkamala, "Towards blockchain-IoT based shared mobility: car-sharing and leasing as a case study," *Journal of Network and Computer Applications*, vol. 200, Apr. 2022, doi: 10.1016/j.jnca.2021.103316.
- [31] D.-C. Huang, L.-C. Liu, Y.-Y. Deng, and C.-L. Chen, "An artwork rental system based on Blockchain technology," *Symmetry*, vol. 15, no. 2, Jan. 2023, doi: 10.3390/sym15020341.
- [32] J. Hu, D. He, Q. Zhao, and K.-K. R. Choo, "Parking management: a Blockchain-based privacy-preserving system," *IEEE Consumer Electronics Magazine*, vol. 8, no. 4, pp. 45–49, Jul. 2019, doi: 10.1109/mce.2019.2905490.
- [33] A. S. Proença, T. R. Dias, and M. P. Correia, "Blockchain based residential smart rent," *arXiv preprint arXiv:2402.05737*, Feb. 2024.
- [34] P. Yadav, S. Sharma, A. Muzumdar, C. Modi, and C. Vyjayanthi, "Designing a trustworthy and secured house rental system using





- Blockchain and smart contracts,” in *2022 IEEE 19th India Council International Conference (INDICON)*, Nov. 2022, pp. 1–6, doi: 10.1109/INDICON56171.2022.10039764.
- [35] Q. Xue, Z. Hou, H. Ma, H. Zhu, X. Ju, and Y. Sun, “Housing rental system based on blockchain Technology,” *Journal of Physics: Conference Series*, vol. 1948, no. 1, p. 12058, Jun. 2021, doi: 10.1088/1742-6596/1948/1/012058.
- [36] S. Gatt and F. Inguanez, “Use of Blockchain technology in automation of Ad-Hoc leasing agreements,” in *2021 IEEE 11th International Conference on Consumer Electronics (ICCE-Berlin)*, Nov. 2021, pp. 1–6, doi: 10.1109/ICCE-Berlin53567.2021.9720013.
- [37] H. Pallevada, G. P. K. Kanuri, S. Posina, S. Paruchuri, and M. Chinta, “Blockchain based decentralized vehicle booking service,” in *2021 2nd International Conference on Smart Electronics and Communication (ICOSEC)*, Oct. 2021, pp. 1418–1424, doi: 10.1109/ICOSEC51865.2021.9591711.
- [38] T.-C. Kang, C.-H. Chang, Y.-W. Chan, Y.-T. Tsai, and T.-E. Liu, “An implementation of house rental platform with Blockchain technology,” in *Frontier Computing*, Springer Singapore, 2020, pp. 298–301.
- [39] L. Wang, X. Lin, E. Zima, and C. Ma, “Towards Airbnb-like privacy-enhanced private parking spot sharing based on Blockchain,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 2411–2423, Mar. 2020, doi: 10.1109/tvt.2020.2964526.
- [40] M. Li, X. Dong, Z. Cao, and J. Shen, “PPHR: Blockchain-based privacy protection house rental system,” in *2021 2nd International Conference on Computer Communication and Network Security (CCNS)*, Jul. 2021, pp. 145–149, doi: 10.1109/CCNS53852.2021.00035.
- [41] Q. Zhou, Z. Zeng, K. Wang, and M. Chen, “Privacy protection scheme for the internet of vehicles based on private set intersection,” *Cryptography*, vol. 6, no. 4, Dec. 2022, doi: 10.3390/cryptography6040064.
- [42] P. Labs, “IPFS powers the distributed web.” <https://ipfs.tech/> (accessed Feb. 15, 2024).
- [43] ZAMA, “Private smart contracts using homomorphic encryption.” <https://www.zama.ai/post/private-smart-contracts-using-homomorphic-encryption> (accessed Feb. 10, 2024).
- [44] DSRGBRIN, “Privacy-preserving Blockchain reservation model,” *GitHub*, [https://github.com/DSRGBRIN/PP\\_RM\\_PBC](https://github.com/DSRGBRIN/PP_RM_PBC) (accessed Feb. 19, 2024).

## BIOGRAPHIES OF AUTHORS







**Akbari Indra Basuki**     earned M.Sc. from Bandung Institute of Technology (ITB), focusing in computer engineering. He works as a researcher at the National Research and Innovation Agency's Research Center for Artificial Intelligence and Cyber Security (BRIN). His research interests include data security, blockchain, network security, and programmable networks. He actively participates in international conferences that are indexed by IEEE and ACM. He was a KMUTT-Thailand visiting researcher. Additionally, he taught at the University of Padjadjaran as a visiting lecturer. He can be reached via email: akba002@brin.go.id.






**Didi Rosiyadi**     is a research professor at BRIN's Research Center for Artificial Intelligence and Cybersecurity. He earned a Ph.D. in information security from National Taiwan University of Science and Technology. His research focuses on information systems, information security, multimedia watermarking, and blockchain. He has been a visiting researcher at numerous research centers and universities throughout the world, including CERN in Switzerland and The University of Zagreb in Croatia, as well as universities in Italy, Taiwan, Thailand, and Bangladesh. He can be reached via email: didi.rosiyadi@brin.go.id.






**Hadi Susanto**     received a B.Eng. in industrial engineering from Institut Teknologi Bandung (ITB) in Indonesia, as well as an M.B.A. and Ph.D. in industrial management from the National Taiwan University of Science and Technology. He is currently a Lecturer in Industrial Engineering at Telkom University in Indonesia, and he is collaborating on research with the National Research and Innovation Agency (BRIN). His research interests include supply chain management, optimization, shelf space allocation, routing issues, and waste management. He has already published his work at a number of conferences, including the Asia Pacific Industrial Engineering and Management Society (APIEMS) and Operations Research and Maritime Logistics. He can be contacted by email: hadist@telkomuniversity.ac.id.



**Iwan Setiawan**    received the M.Sc. degree in instrumentation and control from Institut Teknologi Bandung, Indonesia, in 2006. He is currently a researcher with the Data and Information Preservation Group, National Research and Innovation Agency, Indonesia. His research interests include wavelet, graph theory, and their applications in digital image. He can be contacted at email: iwan022@brin.go.id.



**Taufik Ibnu Salim**    is a researcher at Research Center for Smart Mechatronics, National Research and Innovation Agency. He received a magister of science from Bandung Institute of Technology, majoring in instrumentation and control. His research interests are related to instrumentation, IoT, embedded machine learning and autonomous electric vehicles. He was freelance lecturer at UIN Sunan Kalijaga, Indonesia. He can be contacted at email: tauf021@brin.go.id.