2022 6th International Conference on Cryptography, Security and Privacy (CSP) | 978-1-6654-7975-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/CSP55486.2022.00036

Blockchain-based Smart Parking System using Ring Learning With Errors based Signature

Jihan Lailatul Atiqoh
School of Computing
Telkom University
Bandung, Indonesia
Jihanlailatula@student.telkomuniversity.ac.id

Ari Moesrami Barmawi
School of Computing
Telkom University
Bandung, Indonesia
mbarmawi@melsa.net.id

Farah Afianti
School of Computing
Telkom University
Bandung, Indonesia
farahafi@telkomuniversity.ac.id

***Abstract*—Recently, placing vehicles in the parking area is becoming a problem. A smart parking system is proposed to solve the problem. Most smart parking systems have a centralized system, wherein that type of system is at-risk of single-point failure that can affect the whole system. To overcome the weakness of the centralized system, the most popular mechanism that researchers proposed is blockchain. If there is no mechanism implemented in the blockchain to verify the authenticity of every transaction, then the system is not secure against impersonation attacks. This study combines blockchain mechanism with Ring Learning With Errors (RLWE) based digital signature for securing the scheme against impersonation and double-spending attacks. RLWE was first proposed by Lyubashevsky et al. This scheme is a development from the previous scheme Learning with Error or LWE.**

Keywords—smart parking system; blockchain; digital signature; ring LWE.

I. Introduction

Vehicle parking placement has been a common problem in metro cities, mostly in their malls. An increase in air pollution also follows because of how long it takes for the driver only to drive around the area [3]. With the awareness of the problem, smart parking system ideas start to surface around researchers. The system's goal is to help drivers to find available empty spots to occupy without having to drive around the parking area [1].

The proposed system is cloud computation based to process raw data, web API application to process the data, mobile application for the users to interact with the system, and database to store parking spots information and user data [2].

Regardless of the benefits gained by the smart parking system, there are also some security challenges it has to face. Most smart parking systems have a centralized system, wherein that type of system is generally vulnerable [3] and especially a centralized system is at risk of single-point failure that can affect the whole system [4].

To overcome the risk of single-point failure that can affect the whole system of the centralized system, the most popular mechanism that researchers proposed is blockchain. The reason is blockchain technology is known as a decentralized model that also gives assurance to the data integrity [4]. In 2020 Waheed A. proposed the use of biometric framework and blockchain on smart parking systems [1].

The previous method only explains the use of blockchain in the smart parking system, the mechanism to verify the authenticity of every transaction has been undiscussed. In this paper, the blockchain mechanism is combined with RLWE based digital signature for verifying the authenticity of every transaction. Ring Learning With Errors was first proposed by Lyubashevsky et al. [6]. Lattice-based cryptography schemes are the most promising family in quantum-resistant schemes because of their versatility and superior performance [9]. Hyeongcheol et al. stated that RLWE based digital signatures have a smaller signature size and smallest time complexity compared with other quantum-resistant cryptography [10].

II. The Previous Method

Waheed et al. [1] did a study comparing blockchain and biometric security mechanisms in the smart parking system. Waheed's smart parking system is shown in Fig 1. The result is that blockchain can be used as decentralized storage that can assure security, transparency, and availability of the data system. While biometric is used for accessing the parking system.

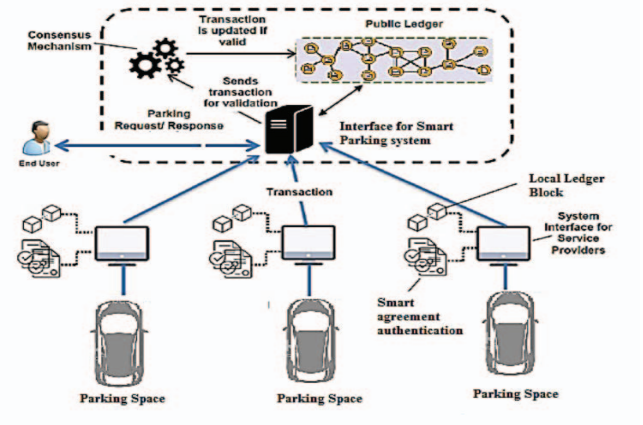

Fig. 1. Smart Parking System overview proposed by Waheed et al. [1]

Waheed et al. did not explain a mechanism to verify the authenticity of the transactions. By assuming that there is no

978-1-6654-7975-2/22/$31.00 ©2022 IEEE
DOI 10.1109/CSP55486.2022.00036

such mechanism, the system is not secure against impersonation attacks, as well as double-spending attacks.

III. RLWE BASED DIGITAL SIGNATURE

Ring Learning With Errors or Ring-LWE is a derivation of LWE that was introduced the first time by Lyubashevsky et al. [5] which defined A as vector *a* in the ring of $\mathbb{Z}[x]/(x_n+1)$. Polynomial multiplication is continuously repeated in the RLWE process, it is computationally demanding, most of the time polynomial multiplication is implemented by using the Number Theoretic Transform (NTT), which reduces the time complexity from O(n^2) to O(*n log n*) [8]. RLWE also has a benefit where it is secure against a cryptanalytic attack by a quantum computer [7].

In 2012 the Lyubashevsky et al. [5] also publish the RLWE based signature [6]. The RLWE based digital signature scheme contains 3 different processes: key generation, signature generation, signature verification.

A. Notation

To understand the process of key generation, at first we have to find a prime number *p* which is congruent to 1 modulo 2*n*, *n* is an integer that is power of 2, and R^{p^n} is the ring of $\mathbb{Z}[x]/(x_n+1)$. Each element in R^{p^n} is a representation of polynomial of degree *n*-1 that has coefficients in range of $[-(p-1)/2,(p-1)/2]$. Then R^{p^n} has a subset named $R_k^{p^n}$ that consist of coefficients in range $[-k,k]$, where $k<\sqrt{p}$.

B. Hash Function

This signature uses a hash function, and the output of this function is a polynomial of degree *n*-1 that has all zero coefficients except at most 32 coefficients that are ± 1. To map the coefficients, we first have to split the {0,1}* input into 160-bits string. To make the string into a 512-bits output, we have to observe into the 5-bits string $r_0r_1r_2r_3r_4$ at a time and transforms them into 16-digit string *s*. If r_0 is 0 then insert -1 into the bit position that represented by $r_1r_2r_3r_4$ in string *s*, otherwise 1 is inserted into the bit position that represented by $r_1r_2r_3r_4$. For example, if the 5-bits are 10110. Then the value of r_0 is 1. Since $r_1r_2r_3r_4$ is 0110 which represents 6, then, 1 is inserted into the 6^{th} bit position of 16-digits string such that it is 0000001000000000.

C. Key Generation

The secret key (*SK*) consist of secret and error (*s*,*e*). Each variable *s* and *e* are uniformly random chosen from $R_1^{p^n}$.

If the public parameter that is uniformly random chosen from R^{p^n} is *a*, then the public key (*PK*) is *as*+*e*.

D. Signature Generation

In the proposed method, signature is used for identifying whether the transaction is done by a legitimate user or not. Thus, a user has to signed using *SK*, before the it is stored in the blockchain. The input of a signature function is the user's *SK*, public parameter *a*, and transaction message *t*. The signature generation process starts by generating the values of y_1 and y_2 from $R_k^{p^n}$, then hashing ay_1+y_2 along with the transaction information as *c*, and computing sc+y_1 and ec+y_2 as z_1 and z_2.

The signature is represented as a tuple of (z_1,z_2, *c*) where z_1, z_2 has to be a subset of $R_{k-32}^{p^n}$, otherwise the signature is rejected and new values of y_1 and y_2 has to be generated. The new values of y_1 and y_2 are used to calculate new *c*, z_1, and z_2 values. The signature sampling was done, if the value of *k* is too small such that z_1, z_2 are not be in $R_{k-32}^{p^n}$, or if *k* is too large then it will be easy to forge the messages [5]. Algorithm 1 shows the signature generation method.

Algorithm 1 Signature Generation

Input Secret Key SK, Public Parameter a, Transaction t
Output Signature

1: $(s,e) \leftarrow$ SK; Check $\leftarrow$ False;
2: **while** $Check \neq True$ **do**
3: $y_1, y_2 \in R_k^{p^n}$; $c \leftarrow$ Hash(ay_1 + y_2, t);
4: $z_1 \leftarrow sc+y_1$; $z_2 \leftarrow ec+y_2$;
5: **if** z_1 and $z_2 \in R_{k-32}^{p^n}$ **then**
6: Check $\leftarrow$ True;
7: **return** (z_1,z_2,c);
8: **end if**
9: **end while**

E. Signature Verification

This method is used to evaluate the authenticity of the signature received by the receiver before the transaction is stored in the ledger. For verifying a signature, the user's public key is necessary. Since in signature generation the value of *k* is already determined then z_1,$z_2 \in R_{k-32}^{p^n}$, otherwise it is a fake signature. Furthermore, the system has to check the hash of (az_1+z_2-*PKc, t*) using the user's PK to verify whether the signature is fake or not. The algorithm for signature verification is shown in Algorithm 1.

Algorithm 2 Signature Verification

Input Public Key PK, Public Parameter a, Signature Transaction t
Output Verified or Not Verified

1: (z_1,z_2,c) $\leftarrow$ Signature; c' $\leftarrow$ Hash(az_1+z_2-*PKc, t*);
2: **if** z_1, $z_2 \in R_{k-32}^{p^n}$ **and** c' = c) **then**
3: **return** Verified;
4: **else**
5: **return** Not Verified;
6: **end if**

IV. THE PROPOSED METHOD

The proposed method consists of five main processes: driver registration, transaction input, transaction payment, parking lot check-in, and parking lot check-out. The RLWE key generation is conducted in the driver registration process. Meanwhile, the RLWE signature generation and verification will be conducted before storing the transaction in the blockchain.

A. Driver Registration

According to the scheme (Fig 2), driver registration is started by entering the driver's information and storing it in *userData* variable. Every userData is distinguished based on their *userId*. A pair of public and secret keys are generated for each driver. The public keys are sent and stored as *userKey* along with their *userId*.

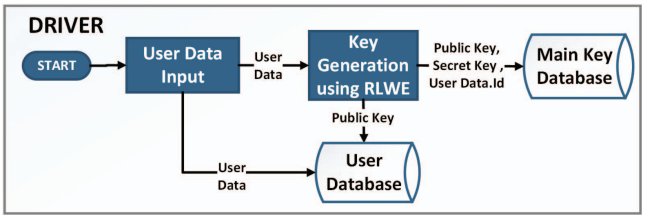

Fig. 2. Driver Registration Scheme

B. Creating Transactions

Creating the transaction process is shown in Fig 3. The main goal of this process is to obtain driver input like the car number license plate, parking spot, entry and out time, etc. when a driver is going to book a parking spot. This information is inserted in a transaction. The transaction has to be signed by the driver using the RLWE method. Furthermore, the signed transaction is sent to the cashier and stored into the blockchain after the signature is verified by the blockchain. If the signature is valid, then the transaction will be added as a new block in the blockchain. Otherwise, it will be suspended.

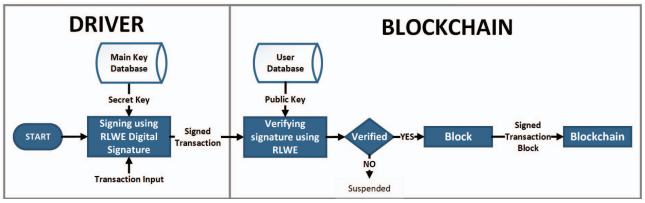

Fig. 3. Transaction Creation Process

C. Transaction Payment

For reserving the parking spot, the driver has to pay the fee first. The payments are made digitally, by assuming that a driver has a unique voucher code with the driver's signature on it.

After entering the transaction, the system will get the transaction id, which is used to identify the transaction details in the blockchain. If the transaction has not been paid yet, then the system will enter the transaction payment process. According to Fig 4, for processing the payment the driver has to pass the voucher code to the cashier through the system. The cashier will check the voucher's availability, the value of the voucher, and its ownership. The ownership is validated using the signature. If the voucher is not valid, then the driver has to put in another voucher code. Otherwise, the transaction voucher can be used and the payment is declared as successful. Fig 5 shows the transaction process.

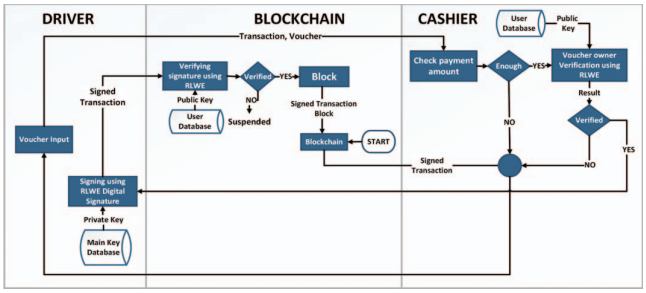

Fig. 4. Transaction Payment Process

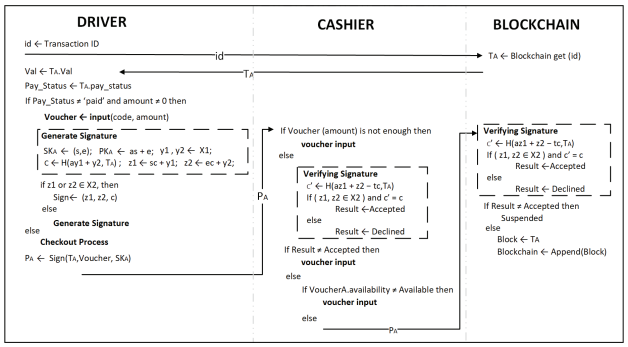

Fig. 5. Payment Scheme

D. Check-in Process

This process is conducted before a driver enters the parking area. The system finds the driver's active transaction and checks the check-in time. If the driver comes to the parking area before the agreed-upon check-in time, he will receive a notification to come again later. If he comes by between the check-in time and check-out time, he can enter the parking area at the parking place he has booked. Then, the transaction status is stored in the blockchain. The overview process is shown in Fig 6.

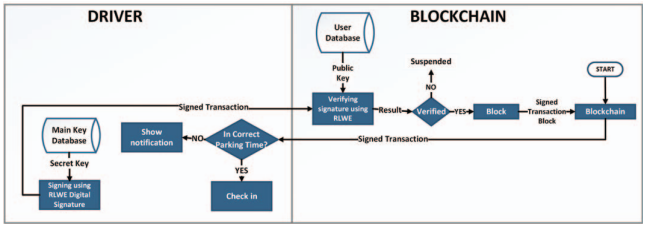

Fig. 6. Check-in Process

E. Check-out Process

This process is conducted before the driver leaves the parking area. As shown in Fig 7, the system checks the

transaction's check-out time and compares it to the current time. If the current time has passed the check-out time, the driver has to pay the overtime parking fee by submitting another voucher code.

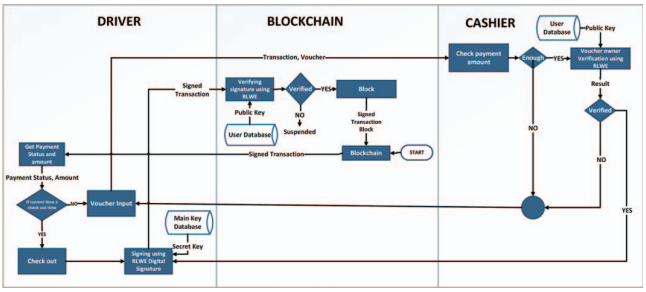

Fig. 7. Check-out Process

V. Discussion

This section discusses the analysis of the proposed method's experiment result, consisting of the performance analysis and the security analysis.

A. Performance Analysis

Performance analysis focuses on the time complexity comparison between the proposed and the previous method. The time complexity of the previous method is O(n) because the algorithm of the system is static, where n is the number of data processed. Since the proposed method uses RLWE based digital signature which relies on polynomial multiplications, then the time complexity of this operation is O($d\ log(d)$) where d is the degree of the polynomials. Furthermore, in the signature generation process, the system has to repeat the signature generation until it meets the requirement where $z1$ and $z1\ z2$ are in $R_{k-32}^{p^n}$. Thus, the time complexity of the entire process is O($2^d * dlog(d) + dlog(d)$).

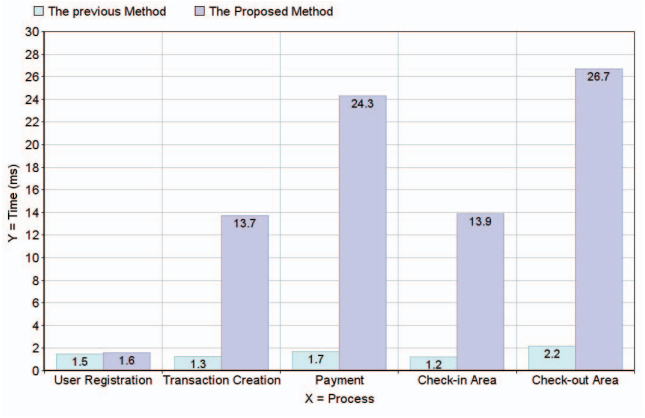

Fig. 8. execution time for each process in seconds

To analyze the execution time, we conduct an experiment of each process using the proposed method as well as the previous one. The experiment result is shown in Fig 8. Based on the experiment result, it is shown that the time used by the proposed method is greater than the time used by the previous method. This condition occurred because, in the proposed method, the RLWE based signature continuously uses polynomial multiplication which has high complexity [8].

B. Security Analysis

In this proposed method, if the attacker succeeds in guessing the secret key, thus the key can be used to create transactions by using the victim's identity. This type of attack is called an impersonation attack. Fig 9 shows how the digital signature prevents someone from impersonation attack by ensuring the authenticity of the driver. In the proposed method, the RLWE based digital signature is also used to ensure the data integrity of the transactions instead of only for the authentication.

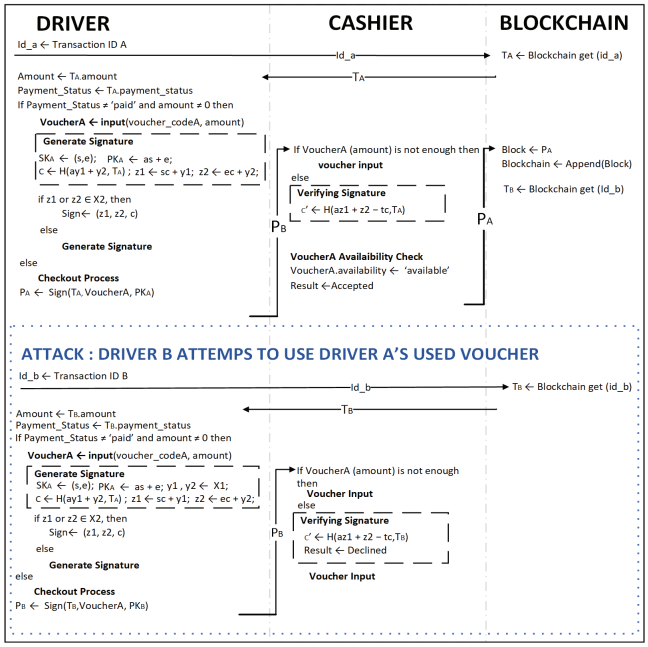

Fig. 9. Impersonation and Double-Spending Attack Scheme

The probability of a successful impersonation attack is equal to the probability of successfully guessing the secret key to produce the digital signature. Since the secret key is represented in tuple *(s,e)* the attacker has to guess each value of s and e. The value of variables s and e are certainly a polynomial and a member of $R_1^{p^n}$. By assuming that the attacker is using a brute-force attack, the probability (p) of guessing the value of each s and e as shown in equation 4.

$$p = 1/c^d \tag{1}$$

where c the amount of number that is the coefficient of the polynomial in the proposed method, the coefficient is in the range of [-1,1] [6], and d is the degree of the polynomial.

$$p = (1/c^d)^2 \tag{2}$$

In addition to the guessing process, the attacker also has to guess the combination of s and e values. Therefore, the attacker has the probability to succeed in the attack (p) as shown in equation 5.

Algorithm 3 Collecting Possible Value of s or e

Input Polynomial Degree d
Output Collection[]

```
q ← 0;
for i_0 = -1 to 1 do
    for i_1 = -1 to 1 do
        ...
        for i_d = -1 to 1 do
            Collection[q] ← [i_0, i_1, ...,i_d];
            q ← q+1 ;
        end for
    end for
end for
return Collection[0..q];
```

Based on Algorithm 3 the time complexity for collecting the possible value of *s* and *e* is O((3^d)). After obtaining every possible combination of *s* and *e*, the attacker has to check the validation of the key by creating a signature using the generated secret key and verifying the signature using the public key.

Algorithm 4 Guess Secret Key

Input public parameter a, Collection[0..q],
Public Key PK, function SignatureGeneration,
function SignatureVerification
Output Valid or Invalid

```
for h = 0 to Collection.length do
    s[h] ← Collection[h]; e[h] ← Collection[h];
end for
t ← Transaction;
for i = 0 to Collection.length do
    for j = 0 to Collection.length do
        SK ← (s[i],e[j]);
        Signed ← SignatureGeneration (SK, a, t);
        Result ← SignatureVerification (PK,a,Signed, t);
        if Result = Valid then
            return Valid;
        end if
    end for
end for
return Invalid;
```

Based on Algorithm 4, the time complexity for conducting an impersonation attack is shown in equation 6.

$$O((3^d) \times (dlog(d)) \times (2^{d+1} + 1)) \tag{3}$$

Finally, guessed keys are utilized in the smart parking system from creating and paying transactions to check-in and check-out.

Meanwhile, by assuming the previous method does not have an authentication mechanism, the time complexity of an impersonation attack in the previous method is equal to the time complexity of its system, which is O(n).

Another attack that is discussed in this scheme is double-spending attacks. Double spending occurs when a voucher is used more than once. As shown in Fig 10, voucher availability is observed. Thus, if a driver is trying to use a used voucher, either if the voucher belongs to him or not, the voucher will not be able to pass the availability observation. This condition occurs because a used voucher is always written in the ledger, and it will be marked as already used. Since the previous method does not include this mechanism, the probability of success double-spending attack is 1, and the probability of success double-spending

VI. Conclusion

Recently, placing vehicles in the parking area is becoming a problem. A smart parking system is proposed to solve this problem. Previously, a centralized parking system is used, but it has a problem that it is at risk of a single-point failure that can affect the whole system. To overcome the centralized problem, a blockchain method using RLWE based digital signature is proposed. The system consists of five main processes: driver registration, transaction creation, transaction payment, check-in area, and check-out area. Based on the analysis, the proposed method has higher computation complexity and execution time compared to the previous method. But, the proposed method is more secure against impersonation and double-spending attacks compared with the centralized parking system.

References

[1] Waheed, Amtul & Krishna, P.. (2020). Comparing Biometric and Blockchain Security Mechanisms in Smart Parking System.
[2] Zajam, Ajay & Dholay, Surekha. (2018). Detecting Efficient Parking Space Using Smart Parking.
[3] Al Amiri, Wesam & Baza, Mohamed & Banawan, Karim & Mahmoud, Mohamed & Alasmary, Waleed & Akkaya, Kemal. (2019). Privacy-Preserving Smart Parking System Using Blockchain and Private Information Retrieval.
[4] Ahmed, Sabbir & ., Soaibuzzaman & Rahman, Mohammad & Rahaman, Mohammad. (2019). A Blockchain-Based Architecture for Integrated Smart Parking Systems.
[5] Lyubashevsky, Vadim & Peikert, Chris & Regev, Oded. (2010). On Ideal Lattices and Learning with Errors over Rings. Journal of the ACM (JACM).
[6] Güneysu, Tim & Lyubashevsky, Vadim & Pöppelmann, Thomas. (2012). Practical Lattice-Based Cryptography: A Signature Scheme for Embedded Systems. System.
[7] Ding, Jintai & Fluhrer, Scott & Rv, Saraswathy. (2018). Complete Attack on RLWE Key Exchange with Reused Keys, Without Signal Leakage.
[8] Nejatollahi, Hamid & Cammarota, Rosario & Dutt, Nikil. (2019). Flexible NTT Accelerators for RLWE Lattice-based Cryptography.
[9] Nejatollahi, Hamid & Shahhosseini, Sina & Cammarota, Rosario & Dutt, Nikil. (2020). Exploring Energy Efficient Quantum-resistant Signal Processing Using Array Processors.
[10] An, Hyeongcheol & Choi, Rakyong & Kim, Kwangjo. (2019). Blockchain-Based Decentralized Key Management System with Quantum Resistance.