IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 13, 2020, accepted May 8, 2020, date of publication May 19, 2020, date of current version June 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2995801

A Lightweight Post-Quantum Lattice-Based RSA for Secure Communications

IQRA MUSTAFA[1], IMRAN ULLAH KHAN[2], SHERAZ ASLAM[3], (Member, IEEE), AHTHASHAM SAJID[4], SYED MUHAMMAD MOHSIN[5], MUHAMMAD AWAIS[6], AND MUHAMMAD BILAL QURESHI[7]

[1]Department of Computing, Cork Institute of Technology (CIT), Cork 021, T12 P928, Ireland
[2]College of Underwater Acoustic Communication Engineering, Harbin Engineering University, Harbin 150001, China
[3]Department of Electrical Engineering, Computer Engineering and Informatics, Cyprus University of Technology, 3036 Limassol, Cyprus
[4]Department of Computer Sciences, BUITEMS, Quetta 87300, Pakistan
[5]Department of Computer Science, COMSATS University Islamabad, Islamabad 45550, Pakistan
[6]School of Computing and Communications, Lancaster University, Lancaster LA1 4YW, U.K.
[7]Department of Computer Science, Shaheed Zulfikar Ali Bhutto Institute of Science and Technology, Islamabad 46000, Pakistan

Corresponding author: Imran Ullah Khan (imrankhann0321@gmail.com)

This work was supported by the Harbin Engineering University, Heilongjiang, China, under Grant HEUCFG201712 and Grant 3072019CFJ0518.

ABSTRACT Conventional RSA algorithm, being a basis for several proposed cryptosystems, has remarkable security laps with respect to confidentiality and integrity over the internet which can be compromised by state-of-the-art attacks, especially, for different types of data generation, transmission, and analysis by IoT applications. This security threat hindrance is considered to be a hard problem to solve on classical computers. However, bringing quantum mechanics into account, the concept no longer holds true. So, this calls out for the modification of the conventional pre-quantum RSA algorithm into a secure post-quantum cryptographic-based RSA technique. In this research, we propose a post-quantum lattice-based RSA (LB-RSA) for IoT-based cloud applications to secure the shared data and information. The proposed work is validated by implementing it in 60-dimensions. The key size is about 1.152×10^5-bits and generation time is 0.8 hours. Furthermore, it has been tested with AVISPA, which confirms security in the presence of an intruder. Moreover, the proposed LB-RSA technique is compared with the existing state-of-the-art techniques. The empirical results advocate that the proposed lattice-based variant is not only safe but beats counterparts in terms of secured data sharing.

INDEX TERMS Lattice-based cryptography, LB-RSA, post-quantum cryptography, IoT application, Gauss-sieve algorithm.

I. INTRODUCTION

Recent years have witnessed that attentiveness towards quantum computing is growing expeditiously due to the extensive need of IoT-based cloud applications for high computation power [1]. The computer experts and professionals have been devising to develop quantum computers. These machines employ the quantum principles, that can utilize quantum mechanical phenomena to solve traditional mathematical problems like 'integer factorization' and 'logarithmic problem', that are incommodious and intractable for classical computers. Also, these devices could power advances in artificial intelligence or easily overwhelm the encryption that protects computers vital to national security.

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

This alarming security situation poses a serious threat to public-key cryptography because it cannot adapt quantum attack by increasing the key-length to outpace the degree of growth of quantum computing. Such security threats would compromise two major aspects of cryptography of digital-communications, especially, when IoT-based cloud applications are considered: confidentiality and integrity over the internet and elsewhere. This security threat hindrance is a great deal of worriment [2], [3].

In 1994, Peter Shor presented the notion of quantum computing; performing physical properties of matter and energy-based calculations and being able to break the RSA cryptosystem within polynomial time whose security relies on integer factorization. Moreover, the publication of Shor's algorithm claims that a powerful quantum computer would be able enough to conquer all sorts of modern techniques

of communication security; from key-exchange to digital authentication of data. So, all the traditional public-key cryptosystems were rendered as impotent [4]. For instance,:

- When quantum computing will become reality, it will become a signal for the end of traditional cryptography [5].
- With the construction of the first quantum-factoring device, the security of the public-key cryptosystem will become extinct [6].

Post-quantum transition raises many fundamental challenges for public-key cryptographic system such as RSA, which need to be addressed to avoid future intimidations. In 2030, quantum computers will be capable of breaking 2000-bit RSA [7]. Moreover, those cryptosystems which are offering 80-bits or less security, which were phased out in $2011 - 2013$, are also at risk [8]–[11]. Even though, the post-quantum RSA's [12] use-cases rely on the faint possibility of considerable improvements in attacks against widespread alternatives and the same criticism is also applied strenuously as discussed in [13].

Hence, there is a need to design a protocol that shows resistance to quantum computers. Among all computational problems that believed to be quantum-safe, lattice-based problems emerged as more economical and quantum-safe encryption providers due to its strong security proof, simplicity, and efficient implementation [14]. In this paper, we have proposed the variant of RSA, which is based on lattices rather than integer factorization problem. In lattice-based cryptography, key selection is not just strong but also hard to break [15]. The private-key for these schemes is a lattice point while the public-key is an arbitrary location in space, which can be the nearest point.

Concluding all, our proposed study has the following contributions:

- Introduction of RSA problem based on lattices rather than integer factorization.
- Security enhancement for communication by increasing key dimensions instead of increasing key-size.
- There are three key facets of our protocol: vector factorization, vector mapping, and finding the shortest vector within n-dimensional lattice.
- Proposed protocol uses the shortest vector problem and vector mapping as a security assumption.
- We have performed a comparison with state-of-the-art algorithms and performed security verification of lattice-based RSA (LB-RSA) using the AVISPA tool.

Initially, this research work implemented the proposed protocol for $n = 60$ dimensions, but to make the protocol resilience against quantum computers, we realized to work in higher dimensions i.e., 100×100. Owing to this fact, we have used matrices for calculating higher dimensions. The proposed scheme motivates researches that rather than increasing the key-length like in post-quantum RSA [12], use the lattices concept i.e., increase security by increasing key dimensions. However, the proposed protocol generates the key of 60-dimension for all types of messages; therefore, this scheme is suitable for long messages.

The rest of the paper is organized as follows. In Section II, we review existing literature, while Section III includes preliminaries. In Section IV, we define methodology along with our new proposed algorithms. In Section V, a discussion about the proposed protocol is given. Security Proof is covered in Section V-B. Section V-C and V-D cover the security analysis and experimental statics of the proposed technique, respectively. Section VI uncovers comparison and contrast with pre-quantum and post-quantum RSA. Finally, Section VII provides conclusion.

II. LITERATURE REVIEW

In the last 30-years, the most widely deployed asymmetric algorithm that provides communication security over a network as discussed above is RSA. Its hardness lies in the integer factorization problem and is considered the most secure algorithm against classical computers [15].

A plethora of research work exists in this domain. In [19], [20] RSA-based security systems have been discussed while in [22], they surveyed three variants designed to speed up RSA decryption. Conventional RSA is believed to be hard for a classical computer to solve [23], but it is not quantum-safe because it cannot adapt quantum attack by enhancing the length of the key to beat the degree of growth of quantum computing.

A. QUANTUM CRYPTOGRAPHY

Quantum computers and quantum cryptography have been extensively discussed in literature [26], [42]. Quantum cryptography devices and methods for communication between two stations are discussed while delivering quantum keys in a single photon is discussed in [28]–[30]. Similarly, David *et al.* [31] discussed Quantum Key Distribution (QKD) protocol for a number of users through a switch. Xu *et al.* [32] suggested that for secure communication, quantum-safe encryption can be achieved using Post Quantum Cryptography and Quantum Cryptography. In another study, [33], the authors have discussed the state-of-the-art advances in Quantum Cryptography, both theoretically and experimentally.

Here we are discussing generally, the schemes whose security lies in the hardness of lattices and how they become resilient to quantum attacks. The purpose of discussing such practical schemes is to bring focus towards the new theories of post-quantum cryptography.

B. LATTICES BASED CRYPTOGRAPHY

Lattice-based cryptography appeared as a better substitute to the existing public-key cryptography because of its quantum-resilience, low key sizes, and versatility. Hamid *et al.* [34] have been well documented the trends in lattice-based cryptography and state-of-the-art applications of lattice adoption in computer security and implementation challenges in software and hardware in their study.

1) NTRU

Later in 1995, Joe Silverman devised a scheme called NTRU which was more efficient than RSA and Diffie-Hellman protocols. This scheme was based on cyclic lattices which were generated by vectors that could rotate in any direction and still land on a lattice point. By 2011, Stehle and Steinfeld proposed ($SS-NTRU$), which is a variant of NTRU encryption scheme, it has reduced the problem to ideal lattices which are closely related to cyclic lattices. These NTRU schemes outperform classical cryptography in terms of performance; however, they have larger key sizes.

2) FULLY HOMOMORPHIC ENCRYPTION (FHE)

In 1997, IBM researcher Cynthia Dwork introduced a first lattice-based scheme, until the worst-case Learning with error (LWE) instances remain computationally hard to be solved. The major difference between classical and present-day encryption scheme is that we don't transform our message instead of noise is added to it. LWE security is based on the hardness of the Shortest Vector Problem (SVP), which requires an efficient quantum algorithm to find it. To hide our message with the error and to avoid computational growth of error, we make our error/message combination small. This proves to be helpful in decryption in a way that if the norm is too high one can find a false point in the lattice and can produce the wrong message. Gentry's Somewhat Homomorphic Encryption (SHE) scheme, which has been improved to FHE scheme through bootstrapping, is based on this concept [35]. However, nowadays, FHE is adopted for various applications, especially for cloud security as a powerful cryptosystem that can carry out computation on encrypted data [36]–[38].

3) RING-LWE

Cryptosystem such as Ring learning with error is also used in practice to boost efficiency however, there exists efficiency-security trade-off. That is because LWE is much versatile and secure than NTRU but not efficient enough. To find the shortest non-zero vector is the core problem in all lattice problems i.e., SVP and contrary to factorization and discrete logarithmic problems no such quantum algorithm exists to solve it. Hence, it is an NP-hard [39].

Motivated by these concepts, in this paper we have proposed a scheme for modifying the conventional integer-based RSA to a LB-RSA, thus help in coping with the future standards of quantum computing and provide a quantum-safe public-key cryptosystem. ''Lattice-based RSA'' public-key cryptosystem could be considered as a strong encryption algorithm replacement of Integer-based RSA.

III. PRELIMINARIES

In this section, different terminologies and definitions related to lattices and prime vectors are discussed. These terminologies are used in later sections of this paper. This section will be quite helpful for the reader to understand the proposed LB-RSA scheme.

A. VECTOR SPACE

A vector space is a set of vectors for which two operations; $+$ and $\times$, are delineated as 'vector-addition' and 'vector-multiplication', respectively. In vector-multiplication, the resultant vector 'C' is known as cross-product or vector-product of the multiplication of two vectors 'A' and 'B' as:

$$C = A \times B \tag{1}$$

where C is a vector product of two vectors A and B

$$C = A \times Bsin\theta \tag{2}$$

The vector-product 'C' is the multiplication-result of vectors' magnitudes time the Sine of included angle as in Equation 2. Besides this, scalar-product; another multiplication-result of two vectors, can be determined by taking a vector's component in the direction of other one.

$$C = A.B \tag{3}$$

where C is a scalar product of two vectors

$$A.B = ABcos\theta \tag{4}$$

In the proposed scheme we have taken two primitive vectors V_1 and V_2; where $V_1 = v_1$ and $V_2 = sv_1$. These vectors are used to construct the product vector N where $N = n_0$. Since vectors are quantities that are described by taking both the magnitude and direction. Each vector has a magnitude and a specific direction. The vectors V_1, V_2 and N have a magnitude as well as direction. In XY-plane we have a maximum angle of $\theta = 180°$. If V_1 has angle θ_1 and the maximum angle is θ then the V_2 has angle θ_2 and it can be calculated as:

$$\begin{aligned} Max\ angle &= \theta \\ Angle\ of\ V_1 &= \theta_1 \\ Angle\ of\ V_2 &= \theta_2 \\ As\ \theta &= \theta_1 + \theta_2 \\ \implies \theta_2 &= \theta - \theta_1 \end{aligned}$$

B. SINGULAR VALUE DECOMPOSITION

As the product of two vectors is either a scalar or a vector. If we calculate the normed vector, we get a scalar value. This value is helpful in finding the pub_{key} and pri_{key} key. The calculated normed value can be used again to find the actual vector. As a single normed can be mapped to different values; for instance, lets first calculate norm of a vector. We can denote the normed vector by $||u||$. The norm of a vector for two dimension can be calculated by:

$$||u|| = \sqrt{u_1^2 + u_2^2} \tag{5}$$

The different normed vectors can be of the same values. If we have two vectors V_1 and V_2. The components of V_1 are $[v_{11}, v_{12}]$ and components of V_2 are $[v'_{21}, v'_{22}]$. If the components of vectors are different, still there is a chance that

normed of different vectors can lead to the same norm. Let's suppose $v_{11} = \alpha$, $v_{12} = \beta$, $v_{21} = \alpha'$ and $v_{22} = \beta'$. The first component of V_1 is equal to the second component of V_2 that is:

$$v_{11} == v_{22} \quad v_{12} == v_{21}$$

The normed of V_1 and V_2 is:

$$\begin{aligned} V_1 &= \sqrt{v_{11}^2 + v_{12}^2} \\ &= \sqrt{(\alpha)^2 + (\beta)^2} \\ &= \sqrt{\gamma^2} \\ &= \gamma \end{aligned}$$

where $\alpha^2+\beta^2 = \gamma^2$. Similarly, to calculate the normed of V_2:

$$\begin{aligned} V_2 &= \sqrt{v_{21}^2 + v_{22}^2} \\ &= \sqrt{\alpha'^2 + \beta'^2} \\ &= \sqrt{\gamma^2} \\ &= \gamma \end{aligned}$$

C. LATTICES

An n-dimensional lattices is the set of all integer combinations of n-linearly independent vectors $\{b_i, \ldots, b_n\} \in R^n$. The set of vectors $\{b_i, \ldots, b_n\}$ is called a basis for the lattice. A basis can be represented by the matrix $B = [b_1, \ldots, b_n] \in R^{n \times n}$ having the basis vectors as columns. The lattice generated by $\mathbf{B}$ is denoted by $\mathcal{L}(B)$. Notice that $\mathcal{L}(B) = \{B_x : x \in Z^n\}$, where $\mathbf{Bx}$ is the usual matrix-vector multiplication [49].

$$\{\sum_{i=1}^{n} x_i b_i : x_i \in Z \, for \, 1 \leq i \leq n\}$$

D. SHORTEST VECTOR PROBLEM

An input to SVP is a lattice $\mathbf{B}$, and the goal is to find a lattice vector of length precisely $\lambda(B)$ [44].

E. PRIMITIVE VECTORS

An n-tuple $[x_1, \ldots, x_n] \in Z^n$ is called primitive iff its coordinates are relatively prime as an n-tuple [45], i.e., [8, 12, 17] is a primitive-vector in Z^3: they are said to be relatively prime as a triple but not pair-wise relatively prime.

F. ONE-WAY FUNCTION

One-way functions are quite simple to compute but it is hard to compute their inverse functions. Hence, having data x it is simple to calculate $f(x)$ while knowing the value of $f(x)$, it is quite hard to find the value of x.

IV. PROPOSED LB-RSA

The proposed LB-RSA algorithm has four subsystems: key-generation, encryption, decryption, digital signing and verification. However, Table 1 shows the symbols that are used in the protocol.

TABLE 1. Symbols and description.

Symbol	Description
$\mathcal{L}(B)$	Lattice-Basis
sv	Shortest-vector
κ	Random Scalar Value
T	Transpose
H	Hash-function
d	Private-key
e	Public-key
I	Identity Matrix
n	No. of Dimensions
σ	Hash Function

A. KEY-GENERATION

The key-generation subsystem uses three distinct primitive vectors from n-dimensional lattice as input and it generates the pub_{key} and pri_{key} as output. Flowchart of LB-RSA Key-Generation is represented by Fig 1. However, steps include in Key-generation are given below:

- Firstly, generate n-dimensional $\mathcal{L}(B)$
- Choose, three random primitive vectors from $\mathcal{L}(B)$, assuming $v_1 = (x_0, x_1, \ldots x_{n-1})$, $v_2 = (y_0, y_1, \ldots y_{n-1})$ and $v_3 = (z_0, z_1, \ldots z_{n-1})$.
- Pass these three n-dimensional vectors where $n = 60$ to Gauss-Sieve (GS) algorithm, which returns a shortest vector SV i.e., $sv_1 = (x', \ldots x'_{n-1})$ which itself is a NP-hard problem where GS algorithm can solve SVP upto 128-dimensions.
- Compute n_0 by performing vector cross product of v_1 and sv_1.
- Calculate Euclidean norm $||x||$ and angle θ from n_0 respectively.
- Compute the totient(ϕ) y of x $[y \leftarrow x]$
- By using angle θ, convert y into vector $c_1 \leftarrow n_1^T$.
- c_1 having 60×60 dimensions, which is our private key d.
- Compute e such that $(\mu)\mu^{-1} = I$, where $\mu = d.\kappa$ and κ is the large random prime scalar value returned by $maxPrime()$ function and I shows the identity matrix.
- Take a message m and convert the message into n-dimensional space and take cross product of $e \times m$ to get m', where m' is the encrypted message.
- In order to decrypt, $m = (\kappa \cdot m') \times d$.

B. ENCRYPTION & DECRYPTION

To perform encryption, take a message $m \in Z_n$ and convert it into $m \times n$ matrix to obtain the ciphertext m' as shown in Alg 2.

If message length is long, sparse it up and encrypt separately. Let e, d, κ be the vectors points $\in Z_n$ with (e) as the encryption and (d, κ) the decryption key, $n = v_1 \times sv_1$. Where, n_0 is public, it will not reveal v_1 and sv_1. Since, the SVP is the basis of security for potentially post-quantum RSA lattice based cyrptosystem. We offer our lattice based sequence for creating a challenge that is able enough to assist determining appropriate sv_1 as SVP for the scheme. Hence, to factor n_0 is NP-hard which assures that d is practically

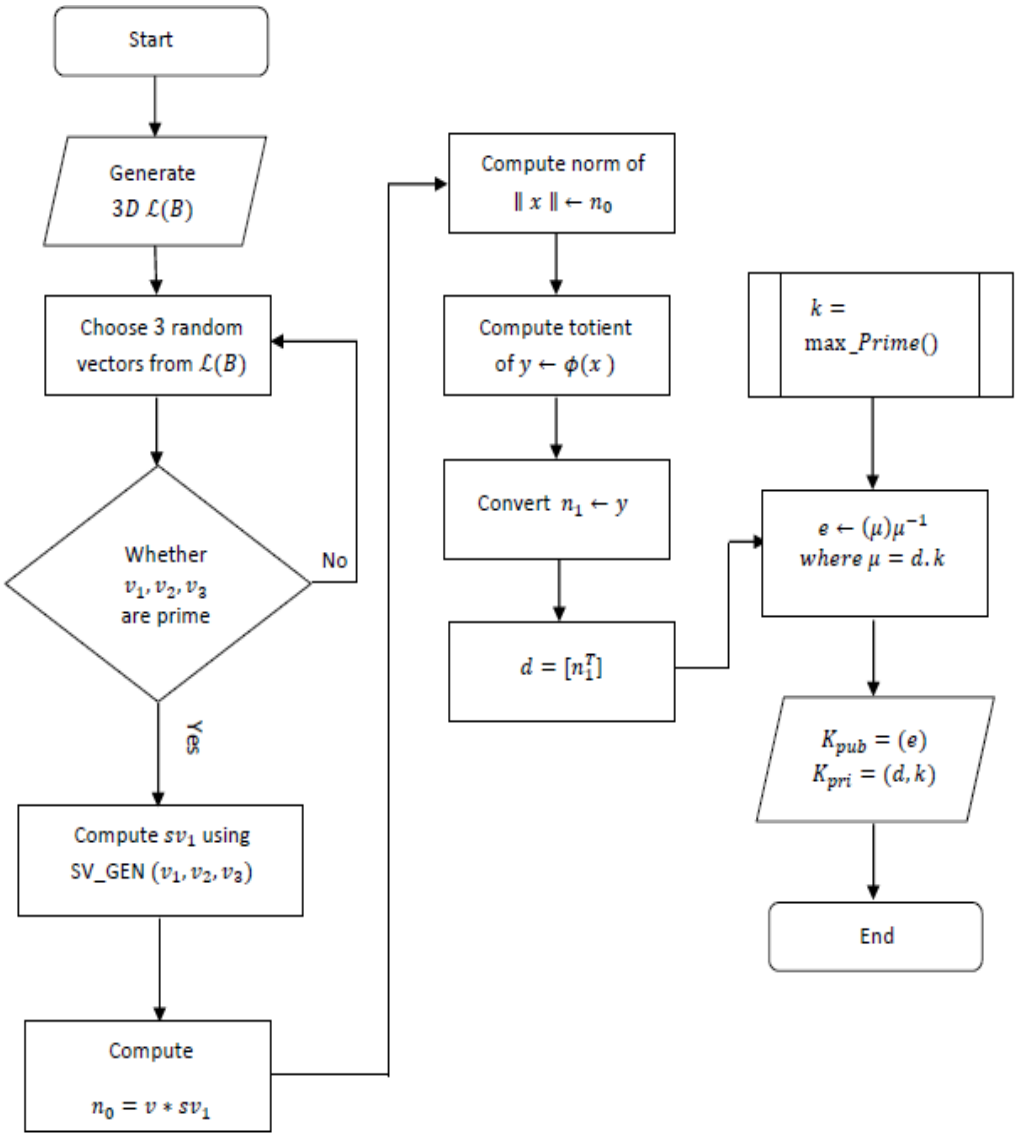

FIGURE 1. Flowchart of LB-RSA key-generation.

impossible to derive from e. For decryption of C_m use equation $m = (\kappa \cdot m') \times d$ in Alg 3.

C. GAUSS SIEVE ALGORITHM

In the proposed protocol, we have used a parallel Gauss-Sieve algorithm [24] in order to find the shortest vector. It was implemented as *gsieve* library by Voulgaris to find the shortest vector sv_1 by passing it sample vector v_1, v_2 and v_3. We used GS as it gives more efficient results as compared to other approaches [24]. In GS (extension) implementation they have solved the SVP Challenge over 128-dimensional lattice, which is currently the highest dimension ever that has been solved.

The Gauss Sieve algorithm comprises a list L of lattice's vectors along with a reduction algorithm giving an output of a shorter vector from two input vectors. The GS algorithm runs a subroutine, Gauss Reduce, which updates v, L, S. Number of collisions of the zero vectors ($\| a' \| = 0$) assist determining the GS algorithm's termination condition that appears in L. The variable K in Alg 4 is the total number of collisions. When the value of K exceeds the threshold condition $\alpha \mid L \mid + \beta$, then the GS algorithm is terminated. In the *gsieve* library, $\alpha = \frac{1}{10}$, and $\beta = 200$ are chosen as the threshold values. The theoretical upper bound of GS algorithm's complexity has not been proved yet. Rather, it outperforms its counterparts in terms of speed.

The key aspect of using this algorithm is that it does not use perturbation, therefore, its space complexity is reduced, and allowing with lattice points only. It builds a list of lattice vectors that are shorter like List Sieve while on an addition to the list of a new-vector v. The Guass Sieve reduced the norm of v using the list vectors. Moreover, it also reduces the

Algorithm 1 Key Generation Algorithm

1:**procedure** KEYGEN
2: **Input**: v_1,v_2,v_3 from $\mathcal{L}(B) = \{b_1, \ldots, b_n\}, \alpha, \beta > 0 \in \mathbb{R}$
3: **Output:** K_{pub},K_{pri}
4: Compute sv_1 using Alg. 4 $SV\_GEN(v_1, v_2, v_3)$
5: Compute $n_0 = v_1 \times sv_1$ where $n \in Z_n^{60\times 60}$
6: Compute $x \leftarrow \| n_0 \|$
7: Compute $x' \leftarrow \phi(x)$
8: Convert x' into $\overrightarrow{n}_1$
9: $c_1 \leftarrow n_1^T$
10: **then** K_{pri} formed is,
11: $d = [c_1]$ column vector having $60 - dimensions$
12: Call *maxPrime*() function to choose κ
13: Perform scalar matrix multiplication of $\kappa \cdot d$
14: Compute e such that $(\mu)\mu^{-1} = I$
15: $K_{pub} \leftarrow (e)$
16: $K_{pri} \leftarrow (d, \kappa)$
17: **end procedure**

Algorithm 2 Ciphertext Generation Algorithm

1: **procedure**: CIPHERTEXT_GEN
2: **Input**: $m \in Z_n$
3: **Output**: $C_m \in Z_n^{60\times 60}$
4: where Z_n is set of integers from 1 to $n-1$
5: Call Algorithm 1 to obtain e
6: Compute $m' \leftarrow m \times e$
7: **end procedure**

Algorithm 3 Plaintext Generation Algorithm

1: **procedure**: PLAINTEXT_GEN
2: **Input**: $C_m \in Z_n^{60\times 60}$ from Algorithm 2
3: **Output**: $m \in Z_n^n$
4: Compute $m = (\kappa \cdot m') \times d$
5: **end procedure**

Algorithm 4 Gauss Sieve Algorithm

1: **procedure**: SV_GEN
2: **Input**:Vectors v_1,v_2,v_3 from Algorithm 1
3; **Output**: SV in $\mathcal{L}(B)$
4: $L \leftarrow \{\}, S \leftarrow \{\}, K \leftarrow 0$
5: **while** $K < \alpha \mid L \mid +\beta$ **do**
6: **if** $S \neq \{\}$**then**
7: pop from stack S to **v**
8: **else**
9: (**v'**, L, S) ←Gauss-Reduce (**v**, L, S)
10: **if** $\| v' \| = 0$ **then**
11: $K \leftarrow K + 1$
12: **else**
13: $L \leftarrow L \cup \{v'\}$
14: **return** a shortest vector in L.
15: **end procedure**

length of those vectors, using vector v that is already in the list. Hence, if min $\{\| v \pm u \|\} <$ max $\{\| v \|,\| u \|\}$ than replace v, u having larger length with shorter $v \pm u$. Hence, list L always contain pairwise reduced vectors.

We made a few changes in GS, to use it with our proposed scheme. However, Alg 4 is a main algorithm of the Gauss Sieve, and Alg 5 and Alg 6 are its subroutines.

Algorithm 5 Gauss Reduce

1: **Input**: Vectors p_1, p_2 in Lattice $\mathcal{L}(B)$
2: **Output**: Vectors $p_1$in lattice $\mathcal{L}(B)$ s.t. $\mid \frac{(p_1,p_2)}{(p_2,p_2)} \mid \leq \frac{1}{2}$
3: **If** $\mid 2.(p_1.p_2) \mid > (p_2.p_2)$ **then**
4: $p_1 \leftarrow p_1 - \lfloor \frac{(p_1,p_2)}{(p_2,p_2)} \rceil . p_2$
5: **return** p_1

Algorithm 6 Gauss Reduce Algorithm

1: **procedure**: GAUSS_REDUCE
2: **Input**: Vector v on$\mathcal{L}(B)$, list L, stack S
3: **Output**: Vector v, list L, stack S, s.t. $v \cup L$ is pairwise reduced
4: *reduce_flag* ← *true*
5: **while** *reduce_flag* = *true* **do**
6: *reduce_flag* ← *false*
7: **for** $l \in L$
8: v' ←Reduce(**v, l**)
9: **if v' ≠v then**
10: *reduce_flag* ← *true*
11: $v \leftarrow v'$
12: **while** $l \in L$ **do**
13: l' ←Reduce(l, v)
14: **if** $l' \neq l$ **then**
15: $S \leftarrow S \cup \{l'\}, L \leftarrow L \backslash \{l\}$
16: **return (v, L, S)**
17: **end procedure**

D. DIGITAL SIGNING & VERIFICATION

For Digital Signing and Verification, we have taken only 3-dimensions. Due to some limitation of the hash function, for instance, if the hash function returns 64-bits, in this case, one needs 264 objects on which that hash function can be called else it won't be collision-free. Hence, we are not considering higher dimension for digital signing. But in future work, we will extend it. We take a file of arbitrary length and compress it into a short string. We used a cryptographic hash function **BlAKE2b**, optimized for 64-bit platforms, and generate digests of any size ranging from 1-64 bytes. In such a way one cannot find n messages that hash to the same value. So, signing the hash value is as good as signing the original message-content without limitation of length. While before generating digest, we pad our message. So that we split it up into multiples of η where η is dependent on our number of dimensions n in which we are working. For 3-dimensions our η will be 9. Let *Alice* is a sender and she performs following steps to sign the message:

- Generate a message digest using cryptographic hash function **BlAKE2b** of the data to be sent.

- Convert $H(m) \in Z_n$ between 1 to $n-1$ called S.
- She uses her $K_{pri}(d, \kappa)$ to compute the signature $S = (H(m) \times \kappa) \times d)$.
- Sends this signature to the recipient *Bob*.

Algorithm 7 Digital Signing Algorithm

1: **procedure**: SIGNING
2: **Input**: $S = H(m)$
3: **Output**: $S_e \in Z_n^{m \times n}$
4: where Z_n is set of integers from 1 to $n-1$
5: Call Algorithm 1 to obtain K_{pri}
6: Compute $S_e = (S \times \kappa) \times d)$
7: **end procedure**

For signature-verification *Bob* perform following steps:

- By using *Alice* $K_{pub} = (e)$ he compute, $\sigma = S_e \times e$
- Compute, independently the message-digest H of the data that has been signed.
- Computes the expected representative σ' by encoding the expected message digest H'.
- If $\sigma = \sigma'$, this depicts signature is valid.

Algorithm 8 Verification Algorithm

1: **procedure**: VERIFICATION
2: **Input**: $S_e \in Z_n^{m \times n}$ from Algorithm 3
3: **Output**: $\sigma = \sigma'$
4: Compute $H' = (s_e \times m)$
5: **end procedure**

V. DISCUSSION, SECURITY PROOF AND SIMULATION RESULTS

This section first presents a detailed discussion about the proposed lattice-based RSA protocol for secure communication and then it uncovers security proof and security verification using AVISPA. In the end, we present a simulation setting and results along with comparative analysis.

A. DISCUSSION ABOUT LB-RSA

We are about to enter the information era, where secure information transmission over the internet is a major concern. Cryptographic protocols are used for this purpose. The existing in-practice cryptographic primitives are either symmetric or asymmetric.

Symmetric primitives have much smaller key-size than the message size and achieve reasonable security in practice but not the perfect security. Besides some symmetric primitives having key-size as long as the message size for one-time-usage achieves perfect security. However, the one-time-usability of such schemes' long keys, puts a question over their applicability. Rather the asymmetric cryptographic primitives have public and private keys.

The public key cryptography primitives are widely deployed in most of the protocols like SSH, OpenPGP, etc., as they achieve confidentiality as well as digital signing function. The hardness of these security primitives lies in integer factorization and logarithmic function etc [53]. Such security primitives would become impotent with the advent of quantum computing [16]. The quantum computers using Shor's and Grover's algorithm would break these algorithms within polynomial time [4], [18]. This serious issue has attained the researchers' attention to have a protocol; resistant enough to quantum computers.

Hence, in the proposed cryptographic protocol, we have used a lattice-based encryption scheme. It is one of the candidates that is considered secure against quantum computers by offering strong security proof, simplicity, and efficient implementation. In lattice-based cryptography, one of the presumed hardness of lattice is the shortest vector problem SVP and in literature various algorithms have been proposed combating this problem. So, one of the security perspectives of our proposed protocol is the hardness of SVP as there is no known polynomial-time algorithm that can be used to solve the exact SVP within polynomial time. The hardness of SVP is discussed in Section V-B3.

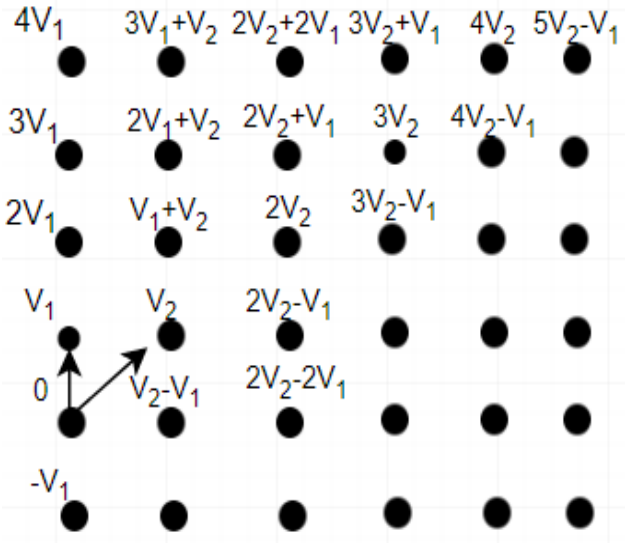

FIGURE 2. Lattices-position.

The given lattice shown in Fig 2 is a linear combination of basis vectors. The span of basis V_1 is the set $\{2V_1, 3V_1, \ldots.\}$ and span of V_2 is the set $\{2V_2, 3V_2, \ldots.\}$. The rest of the vector is a linear combination of the basis vectors. Since all vectors are a linear combination of the rest of the vectors, so the vector formed from any of the given vectors need to be decomposed into the same two vectors. All the basis vectors are mapped to some vector and based on their location, we can complete its decryption process to the original message.

In the given Fig 2, $3V_2$ is the resultant vector of more than one pair of vectors. This vector is in the span of the basis vector and might be a linear combination of the rest of the vectors. A vector pair $[2V_2, 3V_2 - V_1]$ formed the resultant vector $3V_2$; however, this resultant vector may also

be a linear combination of some other vector pair. If during the encryption process this resultant vector is formed from the vector pair $[2V_2, 3V_2 - V_1]$ then it can only be decrypted to the original message if and only if this resultant vector is decomposed into this pair $[2V_2, 3V_2 - V_1]$. The decomposition of the resultant vector into some other vector pair leads us to some garbled value, and decrypting the encrypted message to the original message is not possible.

B. SECURITY PROOF

More importantly, different vectors lead to the same normed value. So, the exact components of a vector cannot be calculated. However, in our proposed scheme, we have taken the vectors from a lattice. Each vector in a lattice holds a specific location. So, vector V_1 and V_2 are different. They have assigned a different location in a lattice. The normed of the vector is unique for each specific position. It means that for each normed there is a unique position and from this norm, we can divert to a specific location. In the above two vectors V_1 and V_2, both have the same normed value. So if we want to find the dimension of the vectors given this normed value, we cannot find the exact vector. Hence, for the proposed scheme we use the concept of vector location.

Given the same normed value for different vectors, we can only locate the exact vector because of the vectors' location. For any two vectors V_1 and V_2, having normed value γ, the decryption of the whole process is only possible when the normed value locate to the exact vector. If γ returns to V_1 (exact location for γ) $\gamma \longrightarrow V_1$. The decryption process returns the exact message m. If γ returns to some other vector $\gamma \longrightarrow (V_2, V_2 \ldots..V_n)$, then the decryption process will return some garbled value instead of m.

Given a set of vectors $V = V_1, V_2, V_3, \ldots\ldots..V_n$, finding the exact vector in V for the given normed value, is based on a number of vectors having the same normed value. Let's suppose we have two vectors V_1 and V_2, whose normed value is the same i.e.,

$$V_1 == V_2 = \gamma$$

where γ is the normed value. The probability (Pb) to find the exact normed value for a given vector is:

$$Pb(V_1) + Pb(V_2) = 1$$

where 1 is the maximum probability. The probability of each vector having the same normed value is equal. Since the maximum probability is 1 and there are only two vectors. The probability of each vector is 0.5.

$$Pb(V_1) = 0.5 \tag{6}$$
$$Pb(V_2) = 0.5 \tag{7}$$

In the case of two vectors, the probability of each vector for a given normed value is half. We are not able to locate the exact vector for given normed value. If we have multiple vectors having the same normed value, then the probability of each vector is further reduced to 1/4. As a result, for a given normed value, the location of the exact vector depends on a number of vectors having the same normed value.

1) INVERSE OPERATOR

Not for all operators, the inverse exists. To check whether it exists or not for an operator, two conditions are need to be checked.

$\tau X \longrightarrow Y$ is called one-to-one mapping of X into Y if and only if $x_1, x_2 \in X$ and $x_1 \neq x_2 \longrightarrow \tau(x_1) \neq \tau(x_1)$ we can say that that τ is one-to-one if inverse of any point $y \in Y$ is at most a single point of X, i.e.,

$$|\tau^{-1}\{y\}| \leq 1 \quad \forall y \in Y \tag{8}$$

$\tau X \longrightarrow Y$ is called on-to mapping if every element of X is mapped to at least one element of y, such as:

$$\tau(X) = Y$$

If $\tau : X \longrightarrow Y$ is one-to-one and onto then an inverse exist for τ denoted by τ_{-1}, such as:

$$\tau(X)y \text{ iff } \tau_{-1}\{y\} = X$$

2) LINEARITY OF INVERSE OPERATOR

If a linear operation $A : X \longrightarrow Y$ (for vectors X and Y) has an inverse, then that inverse A^{-1} is also linear. Suppose

$$A^{-1}(y_1) = x_1 \quad A(x_1) = y_1$$
$$A^{-1}(y_2) = x_2 \quad A(x_2) = y_2$$

then by linearity of A, we have:

$$A(\alpha x_1 + x_2) = (\alpha A x_1 + A x_2) = \alpha y_1 + y_2$$

therefore, $A^{-1}(\alpha A x_1 + A x_2) = \alpha x_1 + x_2 = \alpha A^{-1} y_1 + A^{-1} y_2$.

3) SVP-HARDNESS

There is no known polynomial-time algorithm that is used to solve the exact SVP, as it is an NP-hard problem [49]. The LLL algorithm is the first available algorithm to solve the SVP with running time of $2^{\mathcal{O}(n^2)}$. To solve the exact SVP problem one of the latest algorithms is Discrete Gaussian Sampling that requires 2^n time [51].

4) VECTOR LOCATION

Similarly, RSA is one of the public key encryption schemes and is considered secure against classical computers. Its hardness lies in integer factorization. However, Shor's algorithm can break these cryptographic techniques within polynomial time. Since in our scheme we have discussed the concept of vector factorization and vector location within lattices. Factorizing the vector into two same vectors is one of the difficult problems in lattices as discussed in *"Singular Value Decomposition"*. Vector location is used to solve the problem of vector factorization in lattices. If the same vectors are generated from the resultant vector, then it can lead to a possible solution else it will generate a garbled value, and decryption to the original message is not possible.

C. SECURITY VERIFICATION OF LB-RSA PROTOCOL USING AVISPA

The AVISPA (Automated Validation of Internet Security Protocol and Application) is an extensive tool, that has been designed and used for security protocols' automatic falsification. The protocol falsification refers to the detection of security attacks against the testing protocol unlike the protocol verification; where the correctness of the protocol is more concerned. One of the four backends of AVISPA executes that low-level intermediate code for finding security vulnerabilities in the protocols. Due to its modular approach, AVISPA is a robust tool. There exists a variety of automated falsification tools for security protocols, but the problem is; most of them do not perform well for relatively large-scale security protocols, unfortunately. Alternatively, AVISPA; having a huge library of specification-collections for security protocols and is able enough to specify the large-scale security protocols.

AVISPA provides four back-ends: OFMC, CL-AtSe, SATMC, and TA4SP. OFMC refers to 'On-the-Fly Model Checker'. It takes typed and untyped both protocols models in its consideration. It also provides on-the-fly falsification of protocol and bounded verification. The modeling language used in AVISPA is known as HLPSL. It is used to specify the security properties of the protocols in AVISPA. It is a role-based language in which all participants are represented with some roles.

In our case, we define two roles *Alice* and *Bob*. Our Intruder model is based on two critical aspects that are the perfect encryption and the network (intruder). For this reason, we have formulated two Hypothesis. According to the first claim:

H1: Perfect encryption guarantees that intruder can decrypt m with k if it has the opposite of that key.

H2: Intruder has complete control over the communication channel between participants.

such as, he can modify, block any message passing over the network. We have checked our LB-RSA protocol against OFMC model. The protocol and intruder simulation result shows that our protocol is **"SAFE"** against active and passive attacks including replay and man-in-the-middle attacks as shown in Fig 3 and Fig 4. Due to space limitations, we opted not to add the detail. But LB-RSA.hlpsl code is available on demand.

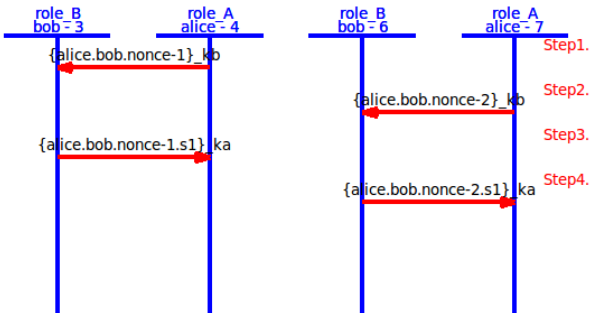

FIGURE 3. LB-RSA protocol simulation using AVISPA.

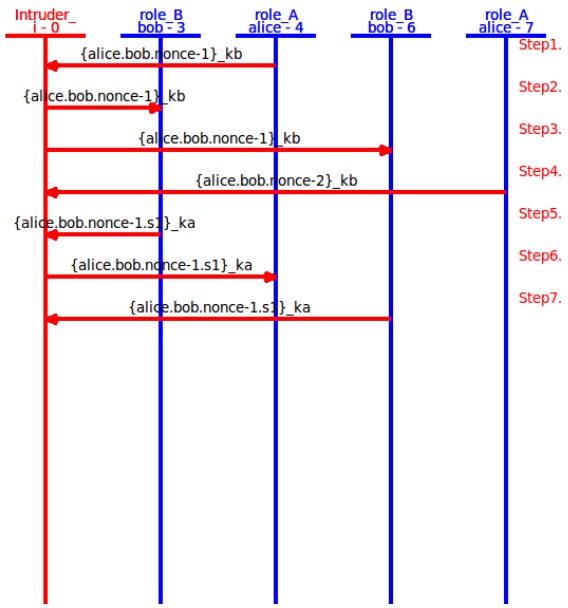

FIGURE 4. Intruder simulation using AVISPA.

D. SIMULATION SETTING

Initially, we have implemented a library based on 60-dimensional lattices. In this approach firstly, we divided the file into chunks i.e., 60×60 matrix. Each chunk is then encrypted. The above process of encryption and decryption is run multiple times as per the number of the chunk. Preliminary results have been drawn for encryption and digital signing. Our key size is, roughly 115200-bits where key-generation time is 0.8 hours for (no. of threads $= 32$).

The computations were carried out on a system running Ubuntu 18.04.3 LTS, (Intel Core i5−8250U), 8GB DDR4 RAM, 256GB SSD. We have implemented our library in Python-3.6 with Komodo Edit 11 IDE and the code requires fpylll for shortest vector computation. Ideally, it should be setup within a python virtual environment. So, we install virtualenv and the pre-requisite packages https://github.com/Iqramustafa293/RSA.

VI. RESULTS AND COMPARISON WITH PRE-QUANTUM AND POST-QUANTUM RSA

With the advent of smart everything and information era, the information appeared as an eminent asset for any organization that needs to be secured through information security measures, i.e., encryption. Up till now, data encrypted under existing schemes were supposed to be secure when transacted across networks. However, with the advent of the latest research in the field of quantum computing, several severe threats have emerged to this supposition. To combat this issue, cryptography researchers toiled to propound ideas of upgrading from simple integer-based methods to lattice-based complex mathematics.

Here we discuss our protocol for 60-dimensional space. Notably, in our proposed scheme, without increasing the key-size ranging from millions of bytes–terabytes, we are

able enough to achieve the same level of security that makes our scheme quantum-safe. The proposed scheme is relatively simple, efficient, and scalable, also it is provably secure under worst-case hardness assumption.

- Like pre-quantum RSA, the security of post-quantum RSA designed by Bernstein *et al.* [12] based on integer-factorization, but in lattices-based RSA we have used the concept of vectors factorization.
- We have generated a lattice basis, and from these, we pick some random vectors to generate a K_{pub} and K_{pri}. So, it is quite impossible for someone to guess the generated lattice basis.
- In LB-RSA we replace the product of 2-large prime integers with a cross-product of 2 n-dimensional vectors. Product of 2 numbers is commutative, while cross-product is non-commutative. It means we can compute the cross-product of 2-vectors but cannot split into 2-independent vectors.
- In post-quantum RSA the authors have generated 1 terabyte exponent-3 RSA key consisting of 4096-bit primes, moreover, the cost of each encryption or decryption takes 1$ of processor time which one would not incorporate in lightweight cryptography. While the key-size of LB-RSA = 14.4 KB. It comes under lightweight cryptography and is suitable for low-cost scenarios.
- In our protocol, it is even easy to compute N but quite impossible to factorize N into 2 independent n-dimensional vectors. Probability to guess each dimension of vector is:

$$Pb(a_{ij}) = 1/n$$

(where n depends on number of vectors $\overrightarrow{n}$ lie in the space, as both vectors v_1 and v_2 are perpendicular to the resulting $\overrightarrow{n}$).

- In vector mapping, we have discussed location at which angle exactly a vector is mapped on, we will have a track on its real predefined angle i.e., the encrypted value is decrypted only when this normed value is factorized into exact vector dimensions, which is difficult for a system to break down.
- LB-RSA key transmission rate and cost is less as compared to post-quantum RSA. A bitter fact about Post-quantum RSA is its computational, storage, and communication cost that makes it highly complex in case of encryption of large contents. Alternatively, our proposed scheme outperforms in terms of simplicity and efficiency.
- The matrix factorization is different from integer factorization, so if one is able enough to break public-key then he must factorize the matrix to guess the security parameter κ. Shor's algorithm is used for integer factorization but does not work for matrix factorization.
- LB-RSA function is indeed a one-way function and is quantum resistive as well with a quadratic attacking cost; because of high dimensions and Singular Value Decomposition of vectors III-B.
- LB-RSA qualifies as secure under archaic security definitions required asymptotic security against polynomial-time adversaries, but it could only be achieved when we increased its dimensions i.e., up to n; where n would represent a larger dimension, providing ultimate as quantum computing becomes reality. So, the current scheme opens the direction towards a path where we don't need to increase the key-lengths to make the algorithms quantum-safe. By changing the security assumptions, we can bring drastic changes in the field of public-key cryptography.

The newly developed LB-RSA efficiency has been tested against [21] and [52] for different file-sizes. Notably, the key-size of LB-RSA is constant for both encryption and decryption. Fig. 5, 6 shows that the encryption and decryption speed of LB-RSA is highly optimized as compared to [21].

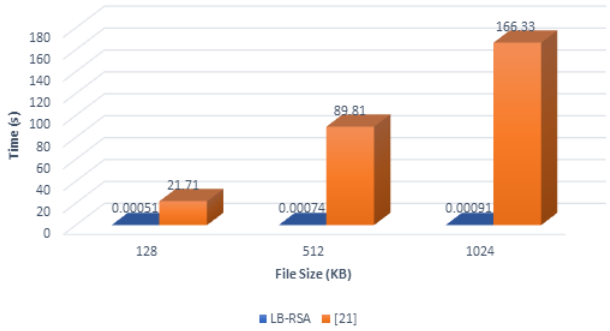

FIGURE 5. Encryption time of LB-RSA and [21].

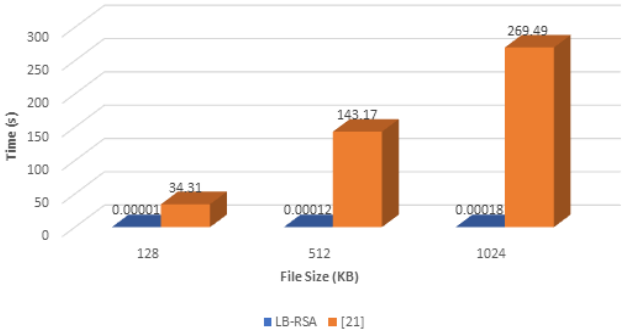

FIGURE 6. Decryption time of & LB-RSA and [21].

Whereas, LB-RSA vs NTRU encryption and decryption time (ms) against different file sizes is illustrated in Fig 7 and 8. Experimental analysis shows that LB-RSA is efficient as compared to NTRU.

Proposed LB-RSA key-generation algorithm runs in exponential time $2^{O(n)}$ and space $2^{O(n)}$ for n-dimensional lattices, it means the time complexity of LB-RSA increases with the increase in number of dimensions. However, the constraint of proposed protocol is the constant key-size i.e., 60-dimensions

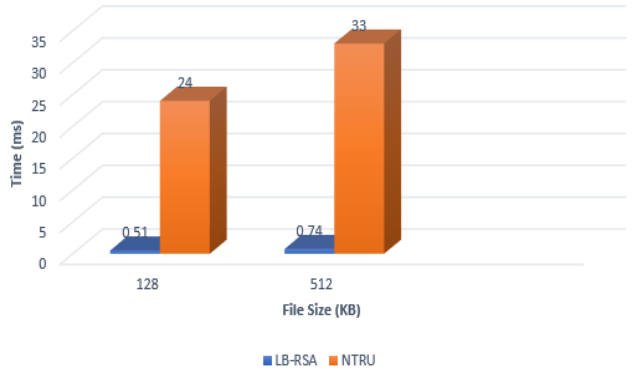

FIGURE 7. Encryption time of LB-RSA and NTRU.

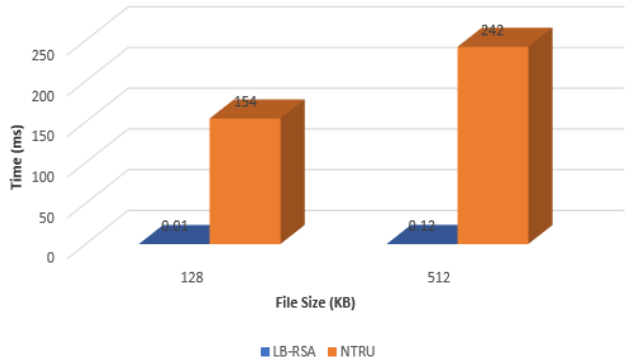

FIGURE 8. Decryption time of & LB-RSA and NTRU.

for all type of messages. Therefore, the adaptability of this scheme is apt for long messages.

VII. CONCLUSION

In this paper, we have presented a novel approach for secure communications by introducing the variant of pre-quantum RSA called lattice-based RSA. The LB-RSA public-key cryptosystem can be considered as a strong encryption algorithm which is the replacement of pre-quantum RSA for IoT-based cloud applications. It is one of the candidates that is considered secure against quantum computers. The reasons for choosing the lattice-based encryption scheme are; the provision of strong security proof for IoT data transmission, simple and efficient implementation for all scenarios, scalability, and efficiency regarding time complexity. Moreover, several security issues could potentially damage the security of integer-based RSA, which are now covered by LB-RSA, such as timing attacks and problems with key distribution.

The comparison of LB-RSA with recent counterparts based on encryption, decryption, key generation time, and total execution time shows that LB-RSA outperforms in terms of operational efficiency and security. Initially, we implemented a library in 60-dimensions for encryption, but for digital signature, we were confined to 3-dimensions only. In the future, we will work on cryptanalysis for digital signing for higher dimensions, e.g., up to 60×60.

REFERENCES

[1] C. Stergiou, K. E. Psannis, B.-G. Kim, and B. Gupta, "Secure integration of IoT and cloud computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 964–975, Jan. 2018.
[2] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Gener. Comput. Syst.*, vol. 82, pp. 395–411, May 2018.
[3] S. Aslam, M. P. Michaelides, and H. Herodotou, "Internet of Ships: A survey on architectures, emerging applications, and challenges," *IEEE Internet Things J.*, early access, May 8, 2020, doi: 10.1109/JIOT.2020.2993411.
[4] P. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM J. Comput.*, vol. 26, no. 5, p. 1484–1509, 1997, doi: 10.1137/s0036144598347011.
[5] S. Boukhonine. (1998). *Cryptography: A Security Tool of the Information Age*. [Online]. Available: https://pdfs.semanticscholar.org
[6] A. Ekert. (2010). *Quantum cryptoanalysis—Introduction*. [Online]. Available: http://www.qi.damtp.cam.ac.uk/node/69
[7] J. Canelán, "A cybersecurity control framework for blockchain ecosystems," Tech. Rep., 2019.
[8] A. Lenstra, E. Tromer, A. Shamir, W. Kortsmit, B. Dodson, J. Hughes, P. Leyland, "Factoring estimates for a 1024-bit RSA modulus," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 2894. Springer-Verlag, 2003, p. 55–74, doi: 10.1007/978-3-540-40061-5-4.
[9] M. Stevens, P. Karpman, and T Peyrin. (2015). *Freestart Collision on Full SHA-1, IACR Cryptology ePrint Archive 2015/967*. [Online]. Available: http://eprint.iacr.org/2015/967
[10] J. Bos, M. Kaihara, T. Kleinjung, A. Lenstra, P. Montgomery. (2009). *On the Security of 1024-Bit RSA and 160-Bit Elliptic Curve Cryptography*. [Online]. Available: http://eprint.iacr.org/2009/389
[11] D. Adrian, K. Bhargavan, Z. Durumeric, P. Gaudry, M. Green, J. A. Halderman, N. Heninger, D. Springall, E. Thomá, L. Valenta, B. VanderSloot, E. Wustrow, S. Zanella-Báguelin, and P. Zimmermann, "Imperfect forward secrecy: How diffie-hellman fails in practice, in *Proc. 22nd ACM Conf. Comput. Commun. Secur.*, Oct. 2015, pp. 1–5. [Online]. Available: http://dx.doi.org/10.1145/2810103.2813707
[12] D. J. Bernstein, "Post-quantum RSA," in *Proc. Int. Workshop Post-Quantum Cryptogr.* Cham, Switzerland: Springer, 2017, pp. 311–329.
[13] G. Brassard, P. Hoyer, K. Kalach, M. Kaplan, S. Laplante, and L. Salvail, "Key establishment àla merkle in a quantum world," 2011, *arXiv:1108.2316*. [Online]. Available: http://arxiv.org/abs/1108.2316
[14] T. Gäneysu, "Getting post-quantum crypto algorithms ready for deployment," Tech. Rep., 2020.
[15] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," *Theor. Comput. Sci.*, vol. 560, pp. 7–11, Dec. 2014.
[16] D. J. Bernstein, J. Buchmann, and E. Dahmen, *Post-Quantum Cryptography*. Springer, 2009.
[17] J. Goldman, "Quantum cryptography—Current methods and technology," Tech. Rep., 2014.
[18] T. Okamoto, K. Tanaka, and S. Uchiyama, "Quantum public-key cryptosystems," in *Advances in Cryptology*. Berlin, Germany: Springer, 2000, pp. 147–165.
[19] A. Gupta, S. Gupta, and N. Yadav, "Enhancement of security using B-RSA algorithm," *Inventive Communication and Computational Technologies*. Singapore: Springer, 2020, pp. 439–450.
[20] I. Mustafa, T. Khan, M. Alam, N. Javaid, A. Khan, and A. Akhunzada, "Post-quantum cryptographic communication protocol," U.S. Patent 15 871 853, Apr. 18, 2019.
[21] P. Gupta, D. K. Verma, and A. K. Singh, "Improving RSA algorithm using multi-threading model for outsourced data security in cloud storage," in *Proc. 8th Int. Conf. Cloud Comput., Data Sci. Eng.*, Jan. 2018, pp. 14–15.
[22] D. Boneh and H. Shacham, "Fast variants of RSA," *CryptoBytes*, vol. 5, no. 1, pp. 1–9, 2002.
[23] M. Campagna, "Quantum safe cryptography and security: An introduction, benefits, enablers and challenges," in *Proc. Eur. Telecommun. Standards Inst.*, 2015, pp. 1–64.
[24] T. Ishiguro, "Parallel gauss sieve algorithm: Solving the SVP challenge over a 128-dimensional ideal lattice," in *Proc. Int. Workshop Public Key Cryptogr.* Berlin, Germany: Springer, 2014, pp. 411–428.
[25] L. Chen, "Report on post-quantum cryptography," US Dept. Commerce, Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep., 2016.
[26] Y. Nambu and A. Tomita, "Quantum cryptography multi-node network system," U.S. Patent 10 184 371, Jun. 28, 2002.

[27] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, "Quantum cryptography," *Rev. Mod. Phys.*, vol. 74, no. 1, p. 145–195, Mar. 2002.
[28] D. S. Bethune and P. William Risk, "An autocompensating fiber-optic quantum cryptography system based on polarization splitting of light," *IEEE J. Quantum Electron.*, vol. 36, no. 3, pp. 340–347, Feb. 2000.
[29] D. B. Patterson and M. J. Kubik, "Free-space quantum cryptography system," U.S. Patent 6 289 104, Sep. 11, 2001.
[30] V. Zimmer and M. Rothman, "Method to support secure network booting using quantum cryptography and quantum key distribution," U.S. Patent 10 940 196, Sep. 13, 2004.
[31] Pearson, David Spencer, and Brig Barnum Elliott, "Simple untrusted network for quantum cryptography," U.S. Patent 7 430 295, Sep. 30, 2008.
[32] F. Xu, X. Ma, Q. Zhang, H.-K. Lo, and J.-W. Pan, "Secure quantum key distribution with realistic devices," 2019, *arXiv:1903.09051*. [Online]. Available: http://arxiv.org/abs/1903.09051
[33] S. Pirandola, U. L. Andersen, L. Banchi, M. Berta, D. Bunandar, R. Colbeck, D. Englund, T. Gehring, C. Lupo, C. Ottaviani, J. Pereira, M. Razavi, J. S. Shaari, M. Tomamichel, V. C. Usenko, G. Vallone, P. Villoresi, and P. Wallden, "Advances in quantum cryptography," 2019, *arXiv:1906.01645*. [Online]. Available: http://arxiv.org/abs/1906.01645
[34] H. Nejatollahi "Post-quantum lattice-based cryptography implementations: A survey," *ACM Comput. Surv.*, vol. 51, no. 6, pp. 1–41, 2014.
[35] B. Lucas, "A brief survey of fully homomorphic encryption, computing on encrypted data," Tech. Rep., 2016.
[36] A. Alabdulatif, I. Khalil, and X. Yi, "Towards secure big data analytic for cloud-enabled applications with fully homomorphic encryption," *J. Parallel Distrib. Comput.*, vol. 137, pp. 192–204, Feb. 2020.
[37] M. Ibtihal and N. Hassan, "Homomorphic encryption as a service for outsourced images in mobile cloud computing environment," *Cryptography: Breakthroughs in Research and Practice*. Hershey, PA, USA: IGI Global, 2020, pp. 316–330.
[38] W. Wang, "Homo-ELM: Fully homomorphic extreme learning machine," in *Proc. Int. J. Mach. Learn. Cybern.*, Jan. 2020, pp. 1–10.
[39] Y. Wang, and Q. M. Malluhi, "Privacy preserving computation in cloud using noise-free fully homomorphic encryption (FHE) schemes," in *Proc. Eur. Symp. Res. Comput. Secur.* Berlin, Germany: Springer, 2016, pp. 301–323.
[40] S. Akleylek, "An efficient lattice-based signature scheme with provably secure instantiation," in *Proc. Int. Conf. Cryptol. Africa*. Cham, Switzerland: Springer, 2016, pp. 44–60.
[41] T. Guneysu, V. Lyubashevsky, and T. Päppelmann, "Practical lattice-based cryptography: A signature scheme for embedded systems," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Berlin, Germany: Springer, 2012, pp. 530–537.
[42] N. Gisin, B. Huttner, A. Muller, H. Zbinden, and B. Perny, "Quantum cryptography device and method," U.S. Patent 6 438 234, Aug. 20, 2002.
[43] D. Micciancio and P. Voulgaris, "Faster exponential time algorithms for the shortest vector problem,"in *Proc. 21st Annu. ACM-SIAM Symp. Discrete, Algorithms*, 2010, pp. 1468–1480.
[44] G. Hanrot, X. Pujol, and D. Stehlá, "Algorithms for the shortest and closest lattice vector problems," in *Proc. Int. Conf. Coding Cryptol.* Berlin, Germany: Springer, 2011, pp. 150–190.
[45] K. Conrad, "Primitive vectors and SLn," Tech. Rep.
[46] A. Terras, "Introduction to normed vector spaces," Tech. Rep., 2009.
[47] O. Christensen, "Normed vector spaces," in *Functions, Spaces, Expansions*, 2010, pp. 29–46.
[48] A. Becker, N. Gama, and A. Joux, "Solving shortest and closest vector problems: The decomposition approach," *IACR Cryptol. ePrint Arch.* vol. 2013, p. 685, Dec. 2013.
[49] D. Micciancio and O. Regev, "Lattice-based cryptography," in *Post-Quantum Cryptography* Berlin, Germany: Springer, 2005, pp. 147–191.
[50] T. Oder, T. Páppelmann, and T. Güneysu, "Beyond ECDSA and RSA: Lattice-based digital signatures on constrained devices," in *Proc. 51st Annu. Design Autom. Conf.*, 2014, pp. 1–6.
[51] D. Aggarwal, D. Dadush, O. Regev, and N. Stephens-Davidowitz, "Solving the shortest vector problem in 2 n time using discrete Gaussian sampling," in *Proc. 47th Annu. ACM Symp. Theory Comput.*, 2015, pp. 733–742.
[52] [Online]. Available: https://github.com/tbuktu/ntru
[53] P. Voulgaris, *Gauss Sieve Beta 0.1, 2010*. San Diego, CA, USA: Voulgaris' homepage at the Univ. California.
[54] F. Bornemann, *Matrix Factorization. In Numerical Linear Algebra*. Cham, Switzerland: Springer, 2018, pp. 21–37.

IQRA MUSTAFA received the master's degree in information security from COMSATS University Islamabad, Pakistan, in 2019. She is currently pursuing the Ph.D. degree in cybersecurity from the Nimbus Research Center, Cork Institute of Technology (CIT), Ireland. She also served as a TPC member and an invited reviewer for international journals and conferences. Her research interests include cryptography, data science, machine learning, generative adversarial networks, blockchain, smart contracts, network security, wireless networks, smart grid, and cloud computing.

IMRAN ULLAH KHAN received the Ph.D. degree in communication networks from the University Malaysia Sarawak (UNIMAS) Malaysia. He has been working as a Professor with the College of Underwater Acoustics Engineering, Harbin Engineering University, China, since September 2018. He has authored many journal articles (SCIE and ESCI) and conference papers. He received research grants from HEC Pakistan as well as Higher Education in Malaysia. Dr. Imran Ullah Khan is a member of the Pakistan Engineering Council (PEC). He is also a Technical Member of National Curriculum Review Committee (NCRC) Higher Education Commission (HEC), Pakistan, a Technical Member of Academic Council (AC), and Board of Study (BOS) the Committee Qurtuba University of Science and IT. He has been serving as a designated reviewer for many reputed international journals.

SHERAZ ASLAM (Member, IEEE) received the B.S. degree in computer science from Bahauddin Zakariya University (BZU), Multan, Pakistan, in 2015, and the M.S. degree in computer science with a specialization in energy optimization in the smart grid from the COMSATS University Islamabad (CUI), Islamabad, Pakistan, in 2018. He is currently pursuing the Ph.D. degree with the DICL Research Laboratory, Cyprus University of Technology (CUT), Limassol, Cyprus, under the supervision of Dr. H. Herodotou, where he is also a part of European Union-funded research project named as STEAM. He worked as a Research Associate with Dr. N. Javaid during his M.S. period at CUI. He has authored more than 20 research publications in ISI-indexed international journals and conferences. His research interests include data analytics, generative adversarial networks, network security, wireless networks, smart grid, cloud computing, and intelligent shipping. He also served as a TPC member and an invited reviewer of international journals and conferences.

AHTHASHAM SAJID is currently serving as an Assistant Professor. He has more than ten years of teaching/ academics experience at various key positions in academics/administrative level at sound known universities in Pakistan. He has five years of industry experience. His research interests include wireless networks, opportunistic networks, the Internet of Things, and under water sensor networks. His research supervision: Supervised 05 M.Phil. Research thesis. His research publications: 03 SCIE, 07 ESCI indexed, 17 HEC recognized journal publications, six international, and four national conference publications.

SYED MUHAMMAD MOHSIN is currently pursuing the Ph.D. degree with COMSATS University Islamabad, Pakistan. His research work appears in impact factor international journals and high ranked national/international conferences. His areas of interest include cyber security, the Internet of Things, edge computing, energy management, and quantum computing.

MUHAMMAD AWAIS received the B.S. degree in computer science from the University of Sargodha, Sargodha, Pakistan, in 2017, and the M.S. degree in computer science from COMSATS University Islamabad, Islamabad, Pakistan, under the supervision of Dr. N. Javaid. He is currently pursuing the Ph.D. degree in communication systems with the School of Computing and Communications, Lancaster University, Lancaster, U.K., under the supervision of Dr. H. Pervaiz. He worked as a Research Associate during his M.S. period in Communications over Sensors Research Group in the Department of Computer Science, COMSATS University Islamabad. He is also working as a Teaching Assistant with Lancaster University. He has authored over more than 15 articles in technical journals and international conferences. He also serves as a regular reviewer for numerous ISI indexed journals. Additionally, he holds the best paper award certificate in an international conference namely EIDWT 2019. He is passionate about Smart Grid, Routing in Underwater Wireless Sensor Networks, the Internet of Things-enabled WSNs, Blockchain-based Systems, Data Science-based WSNs, and the Internet of Things enabled Underwater Sensor Networks.

MUHAMMAD BILAL QURESHI is currently working as an Assistant Professor with SZABIST, Islamabad, Pakistan. He has worked with HPC lab KAU, Saudi Arabia on many funded projects. He is also working on the 2030 Vision project with IAU, KSA. He is the author of many research publications in SCI-E journals, including IEEE Access, the *Journal of Grid Computing*, *Parallel Computing*, the *Journal of Parallel and Distributed Computing*, and *Sustainable Cities and Society*. His research interests include the areas of data-intensive real-time systems, resource allocation problems in HPC systems, and energy-efficient IoT. Dr. Qureshi was a recipient of many prestigious awards, including Gold Medal in undergrad degree, Higher Education Commission Pakistan Indigenous Scholarship for M.S. and Ph.D. studies, research productivity awards, in 2014 and 2015.