Scalar Product Lattice Computation for Efficient Privacy-Preserving Systems

Yogachandran Rahulamathavan, *Member, IEEE*, Safak Dogan, *Senior Member, IEEE*, Xiyu Shi, *Member, IEEE*, Rongxing Lu, *Senior Member, IEEE*, Muttukrishnan Rajarajan, *Senior Member, IEEE*, and Ahmet Kondoz, *Senior Member, IEEE*

***Abstract*—Privacy-preserving (PP) applications allow users to perform online daily actions without leaking sensitive information. The PP scalar product (PPSP) is one of the critical algorithms in many private applications. The state-of-the-art PPSP schemes use either computationally intensive homomorphic (public-key) encryption techniques, such as the Paillier encryption to achieve strong security (i.e., 128 b) or random masking technique to achieve high efficiency for low security. In this article, lattice structures have been exploited to develop an efficient PP system. The proposed scheme is not only efficient in computation as compared to the state-of-the-art but also provides a high degree of security against quantum attacks. Rigorous security and privacy analyses of the proposed scheme have been provided along with a concrete set of parameters to achieve 128-b and 256-b security. Performance analysis shows that the scheme is at least five orders faster than the Paillier schemes and at least twice as faster than the existing randomization technique at 128-b security. Also the proposed scheme requires six-time fewer data compared to the Paillier and randomization-based schemes for communications.**

***Index Terms*—Lattice-based cryptography, privacy-preserving (PP) techniques, scalar product (SP) computation.**

I. Introduction

REGULATORS around the world are enforcing privacy-by-design and privacy-by-default approaches to protect the users' data in rest, transit, and processing. Several service providers and applications that traditionally use users' data in a plain domain to extract patterns and provide services are now applying encrypted-domain computations. Some of the example applications are disease classification in healthcare, data search in the cloud, biometric verification, etc. (e.g., [1]–[8] and references therein). The common theme across these applications is that there are two distrusting parties that want to work on a common goal by combining both of their data while preserving data privacy. For example, a buyer wants to verify his age to an online shop using a security token instead of sending a date of birth.

There are algorithms developed in literature to support data privacy for applications, such as classification algorithms, data mining algorithms, distance calculations, etc. [1]–[8]. In all of these algorithms, one party encrypts the sensitive data whenever that data should be sent to the other party. Hence, the second party needs to process the received data in an encrypted domain. This approach ensures data privacy. Regardless of algorithms, the privacy-preserving scalar product (PPSP) has been used as one of the privacy enabling tools between the two parties. The intuition behind this is that a mathematical function that relies on two different variables can be modified into a scalar product (SP) [3], [4]. Therefore, PPSP becomes a vital tool in most of the privacy-preserving (PP) algorithms.

Suppose, there are two parties, A and B, want to compute the following SP:

$$\mathbf{a}^T\mathbf{b} = \sum_{i=1}^{n} a_i.b_i$$

where vector $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ belongs to A and vector $\mathbf{b} = (b_1, b_2, \ldots, b_n)$ belongs to B. The privacy requirement here is that no party is allowed to learn the other's input vector. At the end, only one party can learn the output of the SP.

Several solutions have been proposed to address this problem in the literature (see Section II). These solutions rely on either public-key encryption techniques to achieve strong security or randomization techniques for high efficiency. The security of these schemes relies on mathematically hard problems and these solutions will be obsolete in a few years' time due to the rise of quantum computers as there are existing quantum algorithms which can easily solve the mathematically intractable problems [9]–[13].

Hence, this article exploits *lattice-based cryptography* to build a PPSP. The proposed model is similar to a lattice-based fully homomorphic encryption scheme [9] and supports multiple encryption and addition without decryption [11]. However, the major challenge was to ensure the error terms are not overflowed to affect the accuracy. This article proposes a methodology to control the error terms while ensuring the given security level, i.e., 128 b.

Lattice-based cryptography has been proven to be secure against quantum attacks and expected to replace the existing

Manuscript received April 4, 2020; revised June 23, 2020; accepted July 24, 2020. Date of publication August 6, 2020; date of current version January 22, 2021. This work was supported by the U.K.–India Education Research Initiative (UKIERI) under Grant UGC-UKIERI-2016-17-019. *(Corresponding author: Yogachandran Rahulamathavan.)*

Yogachandran Rahulamathavan, Safak Dogan, Xiyu Shi, and Ahmet Kondoz are with the Institute for Digital Technologies, Loughborough University London, London E20 3BS, U.K. (e-mail: y.rahulamathavan; s.dogan@lboro.ac.uk; x.shi@lboro.ac.uk; a.kondoz@lboro.ac.uk).

Rongxing Lu is with the Faculty of Computer Science, University of New Brunswick, Fredericton, NB E3B 5A3, Canada (e-mail: rlu1@unb.ca).

Muttukrishnan Rajarajan is with the Information Security Group, School of Engineering and Mathematical Sciences, City University London, London EC1V 0HB, U.K. (e-mail: r.muttukrishnan@city.ac.uk).

Digital Object Identifier 10.1109/JIOT.2020.3014686

public-key cryptography schemes [9]–[13]. Therefore, the proposed solution will be secure against quantum computers and can be used in PP algorithms for various applications to achieve privacy. At the same time, the experimental results (see Section VI) show that the proposed PPSP can also be executed significantly faster than the existing PPSP schemes at the equivalent security level.

The remainder of this article is organized as follows. The related work is discussed in Section II. The background information about lattice-based cryptography and its hardness assumptions are provided in Section III. The proposed algorithm is described in Section IV followed by the security analysis and parameter selections in Section V. Experimental results are provided in Section VI. The conclusions and future work are discussed in Section VII.

II. Literature Review

The existing PPSP schemes can broadly be divided into two: 1) the schemes that are built using proven cryptography such as homomorphic encryption and 2) the schemes that are built based on the information theory, such as randomization and linear algebra. Even though the latter is much efficient than the former, the security level of the latter is not quantified. The following sections study the state-of-the-art algorithms for each of these schemes.

A. Homomorphic Encryption-Based PPSP

Homomorphic encryption techniques such as Paillier play a vital role in supporting PPSP since it offers high security such as 128 b [21]. Even though this scheme is highly secure, it becomes inefficient with the size of the vectors, i.e., it may take a long time (i.e., a few minutes in modern laptops with five cores and 6-GB memory) to compute the SP when the dimension of the vectors is around 1000. Several efficient PPSP schemes were proposed in literature to improve the efficiency [20], [22], [24]–[30]. All these schemes use the homomorphic PPSP scheme as a benchmark to measure efficiency. We discuss these in the following sections.

A lattice-based functional encryption technique that *predicates* whether the SP is equivalent to 0 or not 0 was proposed in [18]. This work is based on lattice trapdoors [16]. If the SP is equivalent to 0, then the trapdoors successfully remove large elements in the problem. Note that the work in [18] is completely different from the objective of the proposed work on this article and the algorithm in [18] cannot be modified to develop a PPSP scheme.

There are works that directly uses learning with errors (LWEs)-based cryptographic scheme for encrypted-domain matrix calculations [34]–[37]. These works treat the encryption technique as a black-box to develop several applications ranging from logistic regression-based prediction to statistics of smart meter reading in an encrypted domain. In contrast to traditional homomorphic encryption such as Paillier, the LWE-based encryption involves a number of parameters that must be set properly for problems with different dimensions. Otherwise, as we will show in Section III, error terms will overflow and decryption will be unsuccessful. In this article, we clearly show how to set up the parameters to achieve a different level of security. Most importantly, this is the first article that compares the performance of quantum-secure cryptographic scheme against traditional homomorphic encryption scheme and information theoretic-secure scheme and shows that a quantum cryptographic-based scheme can outperform the other schemes if the parameters are set properly.

B. Information Theory-Based PPSP

Du and Atallah [24] proposed a PPSP algorithm using 1-out-of-N oblivious transfer function and homomorphic encryption. This algorithm is based on splitting the input vector $\mathbf{a}$ of party A into p number of random vectors to achieve privacy from party B. The drawback of this method is that both parties need to be online and interact several times to perform the SP.

Du and Zhan [25] proposed another SP which reduces the communication complexity of their previous work [24] but with the help of a third-party semi-trusted server. The algorithm in [25] requires a third-party server to generate two random vectors $\mathbf{R}_A$ and $\mathbf{R}_B$. The vector $\mathbf{R}_A$ will be revealed to A and the vector $\mathbf{R}_B$ will be revealed to B. Using these vectors, A and B compute the shares of the SP. Hence, both parties must reveal their shares to get the actual SP value. The communication complexity of this protocol is four times higher than the communication cost of SP without privacy. Moreover, the major drawback of this work is the involvement of a third party who can easily collude with one of the parties to reveal the other party's input vector.

Vaidya and Clifton proposed a novel PPSP solution but without the need of a third party in [26]. The communication complexity of the algorithm in [26] is the same as [25]. However, the computation cost is $O(n^2)$ while it is $O(n)$ for [25]. Moreover, the security of the SP algorithm in [26] depends on the difficulty of solving $n/2$ linear equations.

Amirbekyan and Estivill-Castro [27] proposed a homomorphic encryption and randomization (or the add vector protocol)-based PPSP. Since $2\mathbf{a}^T.\mathbf{b} = \sum_{i=1}^{n} a_i^2 + \sum_{i=1}^{n} b_i^2 - (\mathbf{a}-\mathbf{b})^2$, Amirbekyan and Estivill-Castro [27] exploited homomorphic encryption technique to compute $\mathbf{a}-\mathbf{b}$. Party A generates public-key and private-key pairs using any homomorphic encryption scheme that offers additive homomorphism (i.e., the Pailler encryption) and encrypt the elements of vector $\mathbf{a}$. The encrypted vector and the public key are sent to party B. Party B subtracts its vector $\mathbf{b}$ from encrypted $\mathbf{a}$ using homomorphic properties and obtain encrypted $(\mathbf{a}-\mathbf{b})$. Subsequently, party B permutes and sends the elements of encrypted $(\mathbf{a}-\mathbf{b})$ to party A. Party A decrypts the vector received from party B and obtains the permuted $(\mathbf{a}-\mathbf{b})$. Party A also receives $\sum_{i=1}^{n} b_i^2$ from party B. Using these, party A can compute the required SP. Similarly, there are several variations of PPSP algorithms proposed in literature they either use homomorphic encryption or randomization or both [28]–[30].

One of the algorithms that is secure and lightweight to-date is called secure and PP opportunistic computing proposed in [20] which is proven to be faster than all the other SP and achieve high security. In [20], the security and privacy of the input vectors are protected by masking them by large random

integers whose size is around 512 b. It is shown in [20] that the computational complexity is almost negligible and communication complexity is almost half compared to the Paillier homomorphic encryption-based SP [21]. To make a fair comparison with the proposed scheme, we reset the parameters to achieve 128-b security against traditional computers. Then, in Section VI, we compare the performance of [20] against the proposed lattice-based PPSP scheme and show that the latter one is, at least twice as fast as the [20] algorithm.

Recently, linear algebra-based PPSP was proposed in [22] for biometric identification. The solution proposed is efficient and does not require parties to be online. In particular, the solution is very useful when party A wants to outsource the SP computation to party B.

For this scheme, party A holds both the input vectors $\mathbf{a}$ and $\mathbf{b}$. Initially, party A obtains a diagonal matrix $\mathbf{A}$ using the input vector $\mathbf{a}$ followed by generating two random invertable matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ and a random lower triangular matrix $\mathbf{U}$. The encryption of the input vector $\mathbf{a}$ is simply a matrix multiplication, i.e., $\mathbf{M}_1\mathbf{U}\mathbf{A}\mathbf{M}_2$. This encrypted matrix is send to party B. Later, if party A wants to compute an SP $\mathbf{a}^T\mathbf{b}$, then party A generates a random lower triangular matrix $\mathbf{V}$ and computes $\mathbf{M}_1^{-1}\mathbf{V}\mathbf{B}\mathbf{M}_1^{-1}$ as an encryption of $\mathbf{b}$ where matrix $\mathbf{B}$ is just a diagonal matrix of $\mathbf{b}$. This encrypted matrix is sent to party B who computes the following, which is equivalent to $\mathbf{a}^T\mathbf{b} : \mathbf{Tr}\{\mathbf{M}_1^{-1}\mathbf{V}\mathbf{B}\mathbf{M}_2^{-1}.\mathbf{M}_1\mathbf{U}\mathbf{A}\mathbf{M}_2\}$, where $\mathbf{Tr}$ is a matrix trace operation [19].

This model has been applied in various biometric authentication applications. For example, recently, the work in [23] exploited this scheme to protect biometric templates. In [23], the user extracts biometric template $\mathbf{a}$ and encrypts using random matrices as explained in the previous paragraph. Later, if the user wants to authenticate to the server, then the user extracts a new biometric sample, let us say $\mathbf{b}$, and encrypts using the random matrices and sends it to the server. Using these encrypted samples (i.e., $\mathbf{a}$ and $\mathbf{b}$), the server can find the similarities. This model requires multiplication of several matrices and the complexity will increase substantially when the elements of the matrices are set to large integers to achieve 128-b or higher security. Again, the security of these schemes is dependent on integer factorization and vulnerable for quantum algorithms.

III. Lattice-Based Cryptography

Notations: We use bold lowercase letters like $\mathbf{x}$ to denote column vectors; for row vectors, we use the transpose $\mathbf{x}^T$. We use bold uppercase letters like $\mathbf{A}$ to denote matrices, and identify a matrix with its ordered set of column vectors. We denote horizontal concatenation of vectors and/or matrices using vertical bar, e.g., $[\mathbf{A}|\mathbf{A}.\mathbf{x}]$ where . denotes the matrix multiplication. For any integer $q \geq 2$, we use $\mathbb{Z}_q$ to denote the ring of integers modulo q, $\mathbb{Z}_q^{n\times m}$ to denote the set of $n \times m$ matrix with entries in $\mathbb{Z}_q$. We denote a real number x as $x \in \mathbb{R}$.

A. Lattices

An *m-dimensional lattice* Λ is a full-rank discrete subgroup of $\mathbb{R}^m$ [12]. Let $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ denote the n linearly independent vectors in $\mathbb{R}^m$. Then, m-dimensional lattice Λ is defined to be the set of all integer combinations of $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ as follows:

$$\Lambda = \sum_{i=1}^{n} x_i \mathbf{b}_i \tag{1}$$

where $x_i \in \mathbb{Z}\ \forall i$. The set of vectors $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ is called *basis* for the lattice Λ, and n is called the rank of the lattice.

Without loss of generality, we consider *integer lattices*, i.e., whose points have coordinates in $\mathbb{Z}^m$. Among these lattices, many cryptographic applications use a particular family of the so-called "q-ary" integer lattices which contain $q\mathbb{Z}^m$ as a sublattice for some small integer q. There are two different q-ary lattices considered in many lattice-based cryptographic applications. Let us define them as follows.

1) $\Lambda_q^{\perp}(A)$: For instance, for any integer $q \geq 2$ and any $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, a set of vectors $\mathbf{e} \in \mathbb{Z}^m$ that satisfy the following equation:

$$\mathbf{A}.\mathbf{e} = \mathbf{0} \bmod q \tag{2}$$

forms a lattice of dimension m, which is closed under congruence modulo q. This lattice is denoted by $\Lambda_q^{\perp}(A)$ where

$$\Lambda_q^{\perp}(A) := \left\{\mathbf{e} \in \mathbb{Z}^m | \mathbf{A}.\mathbf{e} = \mathbf{0} \bmod q\right\}. \tag{3}$$

Using $\Lambda_q^{\perp}(A)$, we define a coset or shifted lattice $\Lambda_q^{\mathbf{u}}(A)$ where

$$\begin{aligned}\Lambda_q^{\mathbf{u}}(A) &:= \left\{\mathbf{e} \in \mathbb{Z}^m | \mathbf{A}.\mathbf{e} = \mathbf{u} \bmod q\right\} \\ &= \Lambda_q^{\perp}(A) + \mathbf{x}\end{aligned} \tag{4}$$

where $\mathbf{u} \in \mathbb{Z}_q^n$ is an integer solution to

$$\mathbf{A}.\mathbf{x} = \mathbf{u} \bmod q. \tag{5}$$

2) $\Lambda(A^T)$: Similarly, we can define another m-dimensional *q-ary* lattice, $\Lambda(A^T)$. For a set of vectors $\mathbf{e} \in \mathbb{Z}^m$, and $\mathbf{s} \in \mathbb{Z}_q^n$ which satisfy the following equation:

$$\mathbf{e} = \mathbf{A}^T.\mathbf{s} \bmod q \tag{6}$$

where

$$\Lambda\left(A^T\right) := \left\{\mathbf{e} \in \mathbb{Z}^m | \mathbf{s} \in \mathbb{Z}_q^n \text{ s.t. } \mathbf{e} = \mathbf{A}^T.\mathbf{s} \bmod q\right\}. \tag{7}$$

It is easy to check that $\Lambda_q^{\perp}(A)$ and $\Lambda(A^T)$ are dual lattices.

B. Lattice Hard Problems

There are three well-known hard problems in the lattice that have been exploited by researchers to build several cryptographic applications. This section defines these hard problems briefly.

1) Short Integer Solution: The hardness of finding a short integer solution (SIS) was first exploited by Ajtai [10]. The SIS has served as a foundation for many cryptographic applications, such as the one-way hash function, identification scheme, and digital signature using lattices. The SIS can be defined as follows.

Definition for SIS: For a given m uniformly random vectors $\mathbf{a}_i \in \mathbb{Z}_q^n$, forming columns of a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, finding a

nonzero *short* integer vector $\mathbf{z} \in \mathbb{Z}^m$ with norm $\|\mathbf{z}\| < \beta$ such that

$$\mathbf{Az} = \sum_{i=1}^{m} \mathbf{a}_i.z_i = \mathbf{0} \bmod q$$

is intractable.

This problem has the following useful observations.

1) Without the requirement of $\|\mathbf{z}\| < \beta$, i.e., "short" solution, it is easy to find a vector $\mathbf{z}$ via the Gaussian elimination that satisfies $\mathbf{Az} = \mathbf{0} \bmod q$.
2) The problem becomes easier to solve if m is increased and difficult to solve if n is increased.
3) The norm bound β and the number m of the column vectors must be large enough that a solution is guaranteed to exist. This is the case when $\beta > \sqrt{n.\log(q)}$.

2) Inhomogeneous Short Integer Solution: Inhomogeneous SIS (ISIS) is a variant of SIS. ISIS problem can be defined as follows [11], [12].

Definition for ISIS: For a given m uniformly random vectors $\mathbf{a}_i \in \mathbb{Z}_q^n$, forming columns of a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, and a uniform random vector $\mathbf{u} \in \mathbb{Z}_q^n$, finding a nonzero integer vector $\mathbf{z} \in \mathbb{Z}^m$ with norm $\|\mathbf{z}\| < \beta$ such that

$$\mathbf{Az} = \sum_{i=1}^{m} \mathbf{a}_i.z_i = \mathbf{u} \bmod q$$

is intractable.

3) Learning With Errors: LWEs [9], [13] is an encryption-enabling lattice-based problem but similar to SIS. To enable encryption, the LWE problem depends on a "small" error distribution over integers. The LWE is parametrised by positive integers n and q, and a small error distribution $\mathcal{X} \in \mathbb{Z}_q$, which is typically be a "rounded" normal distribution with mean 0 and standard deviation $(\alpha q/2\pi)$. The constant α plays a critical role in the security of LWE and it should be chosen as large as possible while satisfying the following condition [9]:

$$\alpha q > 2\sqrt{n}. \tag{8}$$

There are two versions of LWE-based problems. Before defining these, let us define a distribution called *LWE-distribution* as follows.

LWE Distribution: For a given *secret* vector $\mathbf{s} \in \mathbb{Z}_q^n$, a sample from LWE distribution $\mathcal{A}_{\mathbf{s},\mathcal{X}} \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ is obtained by choosing a vector $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random, a small error $e \in \mathcal{X}$, and outputting $(\mathbf{a},\ b = \mathbf{s}^T\mathbf{a} + e \bmod q)$.

Using the LWE distribution, we can define two versions of the LWE problem as follows.

1) *Search-LWE:* Given m independent samples $(\mathbf{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ drawn from the above LWE distribution $\mathcal{A}_{\mathbf{s},\mathcal{X}}$ for a uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$ (fixed for all samples), it is intractable to find $\mathbf{s}$.
2) *Decision-LWE:* Given m independent samples $(\mathbf{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ where every sample is distributed according to either: a) $\mathcal{A}_{\mathbf{s},\mathcal{X}}$ for a uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$ (fixed for all samples) or b) the uniform distribution, then distinguishing which is the case is intractable.

We can have the following observations from the two LWE problems outlined above.

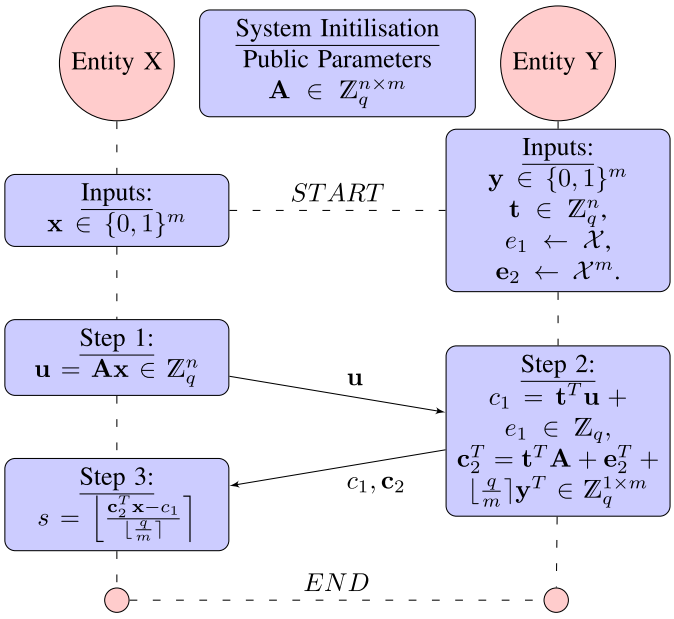

Fig. 1. Flow diagram for the proposed lattice-based PPSP computation for binary vectors.

1) Without the error term $e \in \mathcal{X}$, the search-LWE problem can be solved easily using the Gaussian elimination technique and the secret $\mathbf{s}$ can be recovered.
2) Similarly, for the decision-LWE problem, without the error term $e \in \mathcal{X}$, the Gaussian elimination technique will reveal with high probability that no solution $\mathbf{s}$ exists if it is not sampled from LWE distribution.
3) If there are m LWE samples $(\mathbf{a}_i, b_i) \leftarrow \mathcal{A}_{\mathbf{s},\mathcal{X}}$ for a uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$ (fixed for all samples), we can combine all $\mathbf{a}_i$s into a matrix $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m] \in \mathbb{Z}_q^{n\times m}$, b_is into a vector $\mathbf{b} = [b_1, b_2, \ldots, b_m]^T$, and e_is into a vector $\mathbf{e} = [e_1, e_2, \ldots, e_m]^T$ into the following vector–matrix linear equation:

$$\mathbf{b}^T = \mathbf{s}^T\mathbf{A} + \mathbf{e}^T \pmod q.$$

In the following sections, we will exploit the above lattice hard problems to develop the lattice-based PPSP.

IV. Lattice-Based PP Scalar Product Computation

Let us suppose, there are two distrusting entities, X and Y. Entity X owns an m-dimensional binary vector $\mathbf{x} \in \{0,1\}^m$. Entity Y owns another m-dimensional binary vector $\mathbf{y} \in \{0,1\}^m$. Both X and Y want to interact with each other to compute the SP $s = \mathbf{x}^T\mathbf{y}$ without revealing their own vector to the other party. In the end, one party obtains $s = \mathbf{x}^T\mathbf{y}$. To perform PPSP using lattice, there are four steps required. The following sections describe each of them in detail. The complete algorithm is given in Fig. 1.

1) *System Initialization:* Let us start with generating a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ which is known to X and Y. The matrix $\mathbf{A}$ contains column vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m \in \mathbb{Z}_q^n$, i.e., $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m]$.

2) *Step 1:* Entity X computes an SIS style vector using $\mathbf{A}$ and the binary vector $\mathbf{x}$ as

$$\mathbf{u} = \mathbf{A}\mathbf{x} \pmod q \in \mathbb{Z}_q^n \tag{9}$$

and sends $\mathbf{u}$ to Y.

3) *Step 2:* Entity Y generates a uniformly random vector $\mathbf{t} \in \mathbb{Z}_q^n$, a small error term $e_1 \leftarrow \mathcal{X}$, and a small error vector $\mathbf{e}_2 = [e_{2,1}, e_{2,2}, \ldots, e_{2,m}]^T \leftarrow \mathcal{X}^m$. Then, Y computes the following LWE style term c_1 and vector $\mathbf{c}_2$:

$$c_1 = \mathbf{t}^T\mathbf{u} + e_1 \pmod q \in \mathbb{Z}_q \tag{10}$$

$$\mathbf{c}_2^T = \mathbf{t}^T\mathbf{A} + \mathbf{e}_2^T + \lfloor \frac{q}{m} \rceil \mathbf{y}^T \pmod q \in \mathbb{Z}_q^{1\times m} \tag{11}$$

and sends these to X.

4) *Step 3:* Entity X performs the following computation to retrieve the SP value $s = \mathbf{x}^T\mathbf{y}$ as follows:

$$s = \left\lfloor \frac{\mathbf{c}_2^T\mathbf{x} - c_1}{\lfloor \frac{q}{m} \rceil} \right\rceil. \tag{12}$$

A. Condition for Correctness

Let us derive the condition for the above-mentioned algorithm to output a correct result. In (12)

$$\begin{aligned}\mathbf{c}_2^T\mathbf{x} - c_1 &= \left(\mathbf{t}^T\mathbf{A} + \mathbf{e}_2^T + \left\lfloor \frac{q}{m} \right\rceil \mathbf{y}^T\right)\mathbf{x} - \left(\mathbf{t}^T\mathbf{u} + e_1\right) \\ &= \mathbf{t}^T\mathbf{A}\mathbf{x} + \mathbf{e}_2^T\mathbf{x} + \left\lfloor \frac{q}{m} \right\rceil \mathbf{y}^T\mathbf{x} - \mathbf{t}^T\mathbf{u} - e_1.\end{aligned}$$

Since $\mathbf{A}\mathbf{x} = \mathbf{u}$, and $\mathbf{t}^T\mathbf{A}\mathbf{x} = \mathbf{t}^T\mathbf{u}$

$$\mathbf{c}_2^T\mathbf{x} - c_1 = \left\lfloor \frac{q}{m} \right\rceil \mathbf{y}^T\mathbf{x} + \mathbf{e}_2^T\mathbf{x} - e_1. \tag{13}$$

In (13), the SP is masked by error term $\mathbf{e}_2^T\mathbf{x} - e_1$. To output a correct answer, this error term must satisfy the following condition:

$$\mathbf{e}_2^T\mathbf{x} - e_1 < \left\lfloor \frac{q}{2m} \right\rceil \tag{14}$$

hence

$$\frac{\mathbf{e}_2^T\mathbf{x} - e_1}{\lfloor \frac{q}{m} \rceil} < \frac{1}{2}. \tag{15}$$

Therefore

$$s = \left\lfloor \frac{\mathbf{c}_2^T\mathbf{x} - c_1}{\lfloor \frac{q}{m} \rceil} \right\rceil = \left\lfloor \frac{\lfloor \frac{q}{m} \rceil \mathbf{y}^T\mathbf{x} + \mathbf{e}_2^T\mathbf{x} - e_1}{\lfloor \frac{q}{m} \rceil} \right\rceil = \mathbf{y}^T\mathbf{x}$$

which proves the correctness of the proposed algorithm. Furthermore, the requirements for the error term (14) should be analyzed and defined such that $\mathbf{e}_2^T\mathbf{x} - e_1$ is always smaller than $\lfloor (q/2m) \rceil$. To achieve this, we need to find the upper bound for the error term. The following section is dedicated for this analysis.

B. Upper Bound of the Error Term ($e_2^T x - e_1$)

As we described in Section III-B3, the small error terms are sampled from a normal distribution with mean 0 and standard deviation $(\alpha/\sqrt{2\pi})$ (let us denote this as $\Psi_{0,[\alpha/\sqrt{2\pi}]}$) followed by scaling and modulo reduction by q as follows:

$$e = \lfloor wq \rceil \pmod q \tag{16}$$

where $w \leftarrow \Psi_{0,(\alpha/\sqrt{2\pi})}$ and e belongs to a "rounded" normal distribution with mean 0 and standard deviation $(\alpha q/\sqrt{2\pi})$ (let us denote this as $\mathcal{X}_{0,[\alpha q/\sqrt{2\pi}]}$).

Let us also denote vectors $\mathbf{w} = [w_1, w_2, \ldots, w_m] \leftarrow \Psi_{0,(\alpha/\sqrt{2\pi})}^m$ and $\bar{\mathbf{w}} = [w_1, w_2, \ldots, w_{m+1}] \leftarrow \Psi_{0,(\alpha/\sqrt{2\pi})}^{m+1}$. Hence, the error vector

$$\mathbf{e} = \lfloor \mathbf{w}q \rceil \pmod q. \tag{17}$$

Using the above information, let us find the upper bound for the error term $\mathbf{e}_2^T\mathbf{x} - e_1$. Let us define an $m+1$-dimensional vector $\bar{\mathbf{e}} = [\mathbf{e}_2^T \; e_1]^T$ and another $m+1$ dimensional vector $\bar{\mathbf{x}} = [\mathbf{x}^T \; -1]^T$, hence, $\mathbf{e}_2^T\mathbf{x} - e_1 = \bar{\mathbf{e}}^T\bar{\mathbf{x}}$. Using the triangle inequality, we can define the upper bound of the error term as follows:

$$\left|\mathbf{e}_2^T\mathbf{x} - e_1\right| = \left|\bar{\mathbf{e}}^T\bar{\mathbf{x}}\right| \le \left|(\bar{\mathbf{e}} - q\bar{\mathbf{w}})^T\bar{\mathbf{x}}\right| + \left|(q\bar{\mathbf{w}})^T\bar{\mathbf{x}}\right|. \tag{18}$$

Using the Cauchy–Schwarz inequality [19], we can define the upper bound for the terms in (18) as follows:

$$\left|(\bar{\mathbf{e}} - q\bar{\mathbf{w}})^T\bar{\mathbf{x}}\right| < \|\bar{\mathbf{e}} - q\bar{\mathbf{w}}\|.\|\bar{\mathbf{x}}\| \tag{19}$$

$$\left|(q\bar{\mathbf{w}})^T\bar{\mathbf{x}}\right| < \|q\bar{\mathbf{w}}\|.\|\bar{\mathbf{x}}\|. \tag{20}$$

According to (16) and (17), the rounding error for the components w is at most $(1/2)$ (i.e., $e - \lfloor wq \rceil \le [1/2]$), we have $\|\bar{\mathbf{e}} - q\bar{\mathbf{w}}\| \le (\sqrt{m+1}/2)$ and $\|\mathbf{e}_1 - q\mathbf{w}\| \le (\sqrt{m}/2)$. Hence

$$\|\bar{\mathbf{e}} - q\bar{\mathbf{w}}\|.\|\bar{\mathbf{x}}\| + \|q\bar{\mathbf{w}}\|.\|\bar{\mathbf{x}}\| \le \frac{\sqrt{m+1}}{2}\|\bar{\mathbf{x}}\| + \|q\overline{\mathbf{w}}\|.\|\bar{\mathbf{x}}\|.$$

Since $\bar{\mathbf{x}} \in \{0,1\}^{m+1}$, the Euclidean norm of $\bar{\mathbf{x}}$ is $\|\bar{\mathbf{x}}\| \le \sqrt{m+1}$. Hence

$$\frac{\sqrt{m+1}}{2}\|\bar{\mathbf{x}}\| + \|q\overline{\mathbf{w}}\|.\|\bar{\mathbf{x}}\| \le \frac{m+1}{2} + \|q\overline{\mathbf{w}}\|.\sqrt{m+1}.$$

Since $\overline{\mathbf{w}} \leftarrow \Psi_{0,(\alpha/\sqrt{2\pi})}^{m+1}$ and $q\overline{\mathbf{w}} \leftarrow \mathcal{X}_{0,(q\alpha/\sqrt{2\pi})}^{m+1}$, if we choose standard deviation as 4.5, then the probability

$$Pr\left(|qw| > 4.5 \times \frac{q\alpha}{\sqrt{2\pi}}\right) < 2.5 \times 10^{-7}$$

(i.e., one in four million). The probability will decrease further if we choose a higher number of standard deviations for the upper bound. Without loss of generality, in the rest of this article, we consider standard deviation as 4.5. Therefore, with very high probability

$$\|q\bar{\mathbf{w}}\| \le 4.5q\alpha\sqrt{\frac{m+1}{2\pi}}. \tag{21}$$

Therefore, with very high probability, the error

$$\begin{aligned}\left|\mathbf{e}_2^T\mathbf{x} - e_1\right| &\le \frac{m+1}{2} + \|q\overline{\mathbf{w}}\|.\sqrt{m+1} \\ &\le \frac{m+1}{2} + 4.5q\alpha\sqrt{\frac{m+1}{2\pi}}.\sqrt{m+1}.\end{aligned}$$

As long as this error is smaller than $\lfloor (q/2m) \rceil$, i.e.,

$$\frac{m+1}{2} + 4.5q\alpha\frac{(m+1)}{\sqrt{2\pi}} \leq \left\lfloor \frac{q}{2m} \right\rceil \tag{22}$$

our proposed solution outputs a correct result. Hence, if the upper bound for α is

$$\alpha \leq \frac{\sqrt{2\pi}}{4.5q(m+1)}\left[\left\lfloor \frac{q}{2m} \right\rceil - \frac{m+1}{2}\right] \tag{23}$$

then with high probability (it may not provide correct result one in four million times), the proposed algorithm outputs a correct result. This concludes the proof for correctness. The requirements for the correctness are listed in Table I.

Extending the inputs from {0,1} to integer inputs {0,1,2, ..., 1} will lead to a smaller bin size, i.e., $q/(m * l^2)$. Using this smaller size, (14)–(23) can be revised to obtain parameters for input {0,1,2, ..., 1}. The next section analyzes the security of the proposed algorithm.

V. Security Analysis

As defined in Section IV (refer to Fig. 1), the objective is to protect the privacy of $\mathbf{x}$ from Y and $\mathbf{y}$ from X. Entities X and Y interact with each other to compute the SP.

First, let us prove that Y cannot learn the secret vector $\mathbf{x}$ from the exchanged vector $\mathbf{u}$ in step 1. Since $\mathbf{x} \in \{0, 1\}^m$ (therefore, $\mathbf{x}$ is a short vector), according to the hardness of the ISIS problem defined in Section III-B, it is intractable for Y to solve $\mathbf{u} = \mathbf{A}\mathbf{x} \bmod q$ and obtain a short vector as a solution.

Step 1 operation is similar to hashing. Since the dimension of typical vector $\mathbf{x}$ is 10 000, there are $2^{10\,000}$ possibilities. The only problem is (as same as in any hashing algorithm) the output of step 1 is deterministic for the same $\mathbf{x}$.

Therefore, the brute force approach may not work for Y. Hence, Y needs to use mathematical properties to solve the problem to uncover $\mathbf{x}$ from $\mathbf{u}$. In other words, if Y can recover $\mathbf{x}$ from $\mathbf{u}$, then Y can solve the lattice hardest problem. As defined in Section III-B, Y cannot find a vector $\mathbf{x}$ shorter than β, i.e., $\|\mathbf{x}\| < \beta$. Therefore, let us analyze the shortest possible vector which can be recovered by Y.

Suppose if Y wants to find a short vector $\mathbf{x}$ from $\mathbf{u} = \mathbf{A}\mathbf{x} \bmod q$, then Y may exploit the state-of-the-art techniques called the lattice reduction method [14] and/or combinatorial method [15]. Denote the shortest vector which can be found by these techniques as $\mathbf{x}_s$. It is proven in literature (theoretically and experimentally) [17], that the Euclidean length of $\mathbf{x}_s$ has a lower bound as follows:

$$\|\mathbf{x}_s\| \geq \min\left\{q, 2^{2\sqrt{n.\log(q)\log(\delta)}}\right\} \tag{24}$$

where $\delta \geq 1.01$ [14]. Since the X's secret vector $\mathbf{x} \in \{0, 1\}^m$, the Euclidean length $\|\mathbf{x}\| \leq \sqrt{m}$. Hence, using (24) and assuming q is very large, if

$$\sqrt{m} < 2^{2\sqrt{n.\log(q)\log(\delta)}} \tag{25}$$

then Y cannot recover $\mathbf{x}$ from $\mathbf{u}$. This is the first condition for security. This concludes that if condition (25) is met, then Y cannot recover $\mathbf{x}$ from $\mathbf{u}$. Also, the cost (L) of finding a short binary vector using the techniques described above is defined as [17]

$$L \approx 2^{\frac{m}{2^k}} \tag{26}$$

where k should satisfy the following equation:

$$\frac{2^k}{k+1} \approx \frac{m}{n.\log(q)}. \tag{27}$$

Now, let us focus whether X can recover $\mathbf{y}$ from the messages c_1 and $\mathbf{c}_2$ sent by Y to X in step 2.

According to the definition in Section III-B, if c_1 and $\mathbf{c}_2$ are LWE terms, then it is intractable for X to recover $\mathbf{y}$ since c_1 and $\mathbf{c}_2$ are indistinguishable from uniformly random distribution. If $\mathbf{t}$, $\mathbf{u}$, and $\mathbf{A}$ are uniformly distributed and the error term e_1 and error vector $\mathbf{e}_2$ are sampled from normal distribution with standard deviation greater than $2\sqrt{n}$ as defined in (8), then c_1 and $\mathbf{c}_2$ are uniformly random.

Matrix $\mathbf{A}$ is already a uniformly random matrix. Entity Y can generate uniformly random $\mathbf{t}$, e_1 and $\mathbf{e}_2$. The vector $\mathbf{u}$ sent by X is uniformly random as long as the number of possibilities for $\mathbf{x}$ is larger than $\mathbf{u}$, i.e., $2^m > q^n$ or $m > n.\log(q)$ [17] (this is the second security condition).

Since the dimension of $\mathbf{t}$ is $m > 1$, and the scalar $\mathbf{t}^T\mathbf{u}$ is masked by an error term e_1, the term c_1 is scalar and completely random. Therefore, according to the LWE definition, it is intractable for X to recover the elements of $\mathbf{t}$ from scalar c_1. To analyze $\mathbf{c}_2$, let us denote the ith element of $\mathbf{c}_2$ as $c_{2,i}$ where $c_{2,i} = \mathbf{t}^T\mathbf{a}_i + e_{2,1} + \lfloor (q/2m) \rceil y_i$. In $c_{2,i}$, $\mathbf{t}^T\mathbf{a}_i + e_{2,1}$ is scalar and LWE term, i.e., uniformly random. Similar to the LWE encryption scheme [9], $\mathbf{t}^T\mathbf{a}_i + e_{2,1}$ acts like a one-time pad to hide the message $\lfloor (q/2m) \rceil y_i$. Hence, X cannot recover y_i from $c_{2,i}$ and therefore the proposed scheme is secure. In Section V-A, we show that our parameter choice satisfying (8) (third security condition) is hard and at least equivalent to 128-b security.

In LWE, the noise term plays a major role in determining the hardness [9]. The normal distribution where the error terms are sampled must satisfy (8). The α term must be chosen as the largest possible while satisfying (8) for the hardness of LWE. To quantify the hardness or security level of LWE for a concrete set of parameters, Regev *et al.* exploited the dual lattice in [17, p. 21]. The idea is to find how many operations are required to distinguish an LWE term from a uniform distribution. This is only possible if an adversary can find a short vector on the dual lattice. To this, let us denote a vector $\mathbf{v}$ and denote a short vector in the dual lattice as $\mathbf{w}$. If the vector $\mathbf{v}$ is an LWE vector, then the SP $\mathbf{v}^T\mathbf{w}$ will be an integer [17, p. 22]. If not, then $\mathbf{v}$ is a uniform random vector. Therefore, finding a short vector in dual lattice must be hard. If the standard deviation of the error term $\alpha q/2\pi$ is not bigger than $1/\|\mathbf{w}\|$ then it may be possible to find a short vector in the dual lattice. Therefore, the error term must be bigger than $1/\|\mathbf{w}\|$ for LWE security. This requirement and (24) can now be used to quantify the LWE security.

Now, using the lattice properties, i.e., the length of a shorter vector in dual lattice is equivalent to $1/q$ times the length of a shorter vector in lattice [17, p. 22]. Using this and (24), we can say $\|\mathbf{w}\| \approx (1/q).min\{q, 2^{2\sqrt{n.\log(q)\log(\delta)}}\}$. Therefore,

TABLE I
REQUIREMENTS FOR PARAMETERS TO ACHIEVE 128-B SECURITY AND CORRECTNESS WHEN THE STANDARD DEVIATION IS SET FOR 4.5

	n	m
Correctness	$n \geq 1$	$m \geq 1$
Security	$n.log(q) > 128$	$m \geq nlog(q)$ & $\sqrt{m} < 2^{2\sqrt{nlog(q)log(\delta)}}$
	α	q
Correctness	$\alpha \leq \frac{\sqrt{2\pi}}{4.5q(m+1)} \left[\lfloor \frac{q}{2m} \rceil - \frac{m+1}{2}\right]$	$q > 2m$
Security	$\alpha \geq max\left\{\frac{2\sqrt{n}}{q}, 1.5\sqrt{2\pi}.max\left\{1/q, 2^{-2\sqrt{n.log(q).log(\delta)}}\right\}\right\}$	$q > n$

if error

$$\frac{\alpha q}{\sqrt{2\pi}} >> \frac{1}{\|\mathbf{w}\|} \tag{28}$$

then LWE is hard. By taking 1.5 as factor, we can define the lower bound for α from (28) as follows [17]:

$$\alpha \geq 1.5\sqrt{2\pi}.\max\left\{1/q, 2^{-2\sqrt{n.\log(q).\log(\delta)}}\right\}. \tag{29}$$

The cost of finding a shorter vector is the same as (26). In Section V-A, we show that our parameter choice to satisfy (8) is hard and at least equivalent to 128-b security.

A. Parameter Selection

First, let us obtain the relationship between q and m. Since the maximum possible value for $\mathbf{x}^T\mathbf{y}$ is m, we split q into m parts, i.e., the distance between the consecutive values is $\lfloor (q/m) \rceil$. To obtain a correct result, as shown in (22), half of this distance should be larger to accommodate the error term, i.e., $\lfloor (q/2m) \rceil > 1$ or $q > 2m$. Table I provides the necessary requirements for all the parameters to achieve correctness and security. This table is a summary of requirements derived in the previous sections. Using this table, let us obtain a concrete set of parameters to achieve 128-b security. The same strategy has been used to obtain the parameters for lower security (i.e., 80, and 112 b) and higher security 256 b in Section VI.

To obtain 128-b security, we need to choose our parameters in such a way that the cost (26), $L \approx 2^{(m/2^k)} \geq 2^{128}$. If we choose $k = 2$, then from (27), $m \approx n.\log(q)$. Hence, $L \approx 2^{n.\log(q)} \geq 2^{128}$. Therefore, the security of the solution would be equal to 128 b if $n.\log(q) \approx m \geq 128$. Based on this and other requirements (all are listed in Table I), we are proposing six sets of parameters in Table II to achieve 128-b security. These parameters have been cross validated using the well known LWE estimator [33] (the source code for the LWE estimator, that calculates the security complexity using six different algorithms such as lattice-reduction, dual-lattice attacks etc., is available at https://bitbucket.org/malb/lwe-estimator).

In Table II, parameters n and q play a major role to ensure 128-b security. They are linked as increasing n leading to a small q. These parameters determine the size of matrix $\mathbf{A}$ and the memory requirement. The first four sets are equivalent in terms of memory ($\approx$ 100 MB) while the last two require around 200 and 800 MB, respectively. As shown in the experiments, the running time for the last two are significantly higher and not useful for practical applications. For sets V and VI, the size of q is not decreasing as much as those for the other sets. The security levels for sets V and VI are 187 and 517 b, respectively. The reason is that larger n leads to a larger m, hence, in order to satisfy the error distribution parameter α in (23), the value for q must be set to high. Increasing the value for α will increase the security.

TABLE II
CHOICES FOR THE SECURITY PARAMETERS TO ACHIEVE AT LEAST 128-B SECURITY

SET	n	m $\approx$	q $\approx$	$Security$ $\approx$	$\alpha.q$ (error std. $\approx$)
I	50	2^{15}	2^{570}	2^{128}	2^{538}
II	100	2^{15}	2^{270}	2^{128}	2^{238}
III	250	2^{15}	2^{116}	2^{128}	2^{85}
IV	500	2^{15}	2^{55}	2^{128}	2^{24}
V	1000	2^{15}	2^{39}	2^{187}	2^{7}
VI	2000	2^{16}	2^{41}	2^{517}	2^{7}

VI. EXPERIMENTAL RESULTS

In order to evaluate the proposed LWE-based PPSP scheme, we implemented the algorithm in Java and tested on a 64-b Windows PC with 16-GB RAM and Intel Core i5-4210U CPU at 1.70 GHz. For performance comparison, we also implemented the Paillier homomorphic encryption-based PPSP scheme [21] on the same PC using Java. Additionally, we compared our scheme with one of the most efficient PPSP algorithms in [20]. Our test results show that the proposed LWE-based scheme is significantly faster (at least 10^5 times faster) than the Paillier homomorphic PPSP scheme and at least twice as fast as [20] for the 128-b security.

A. Proposed Lattice-Based PPSP Scheme and Paillier PPSP Scheme

The Paillier cryptosystem [21] is an additively homomorphic public-key encryption scheme. Its provable semantic security is based on the decisional composite residuosity problem: it is mathematically intractable to decide whether an integer z is an n-residue modulo n^2 for some composite n, i.e., whether there exists some $y \in \mathcal{Z}_{n^2}^*$ such that $z = y^n \text{mod} n^2$. Let $n = pq$ where p and q are two large prime numbers. A message $m \in \mathcal{Z}_n$ can be encrypted using the Paillier cryptosystem as $[\![m]\!] = g^m r^n \bmod n^2$ where $g \in \mathcal{Z}_{n^2}^*$ and $r \in \mathcal{Z}_n^*$. For a given encryption $[\![m_1]\!]$ and $[\![m_2]\!]$, an encryption $[\![m_1 + m_2]\!]$ can be obtained as $[\![m_1 + m_2]\!] = [\![m_1]\!][\![m_2]\!]$, and multiplication of an encryption $[\![m_1]\!]$ with a constant α can be computed efficiently

TABLE III
PAILLIER HOMOMORPHIC ENCRYPTION-BASED PPSP [21]

Input by X: $\mathbf{a} = [a_1, \ldots, a_m]^T \in \{0,1\}^m$ and **Y:** $\mathbf{b} = [b_1, \ldots, b_m]^T \in \{0,1\}^m$ **Output to X:** $\mathbf{a}^T\mathbf{b}$
Step 1: X performs the following operations: Generates Paillier public-private key pairs $\{pub, sk\}$, FOR EACH a_i, $i = 1, 2, ..., m$ Computes $E_{pub}(a_i) = [\![a_i]\!]$, END FOR keeps sk, and sends $(pub, E_{pub}(a_1) \ldots E_{pub}(a_m))$ to Y
Step 2: Y executes the following operations Using b_i, $i = 1, 2, ..., m+2$ Computes $E(\mathbf{a}^T\mathbf{b}) = [\![a_1]\!]^{b_1}.[\![a_2]\!]^{b_2} \ldots [\![a_m]\!]^{b_m}$ Sends $E(\mathbf{a}^T\mathbf{b})$ back to X
Step 3: X decrypts and obtains $\mathbf{a}^T\mathbf{b} = D_{sk}(E(\mathbf{a}^T\mathbf{b}))$.

TABLE IV
AVERAGE RUNNING TIME FOR THE PROPOSED AND PAILLIER-BASED PPSP SCHEMES

SET	The Proposed Lattice-based PPSP				Pailler Based PPSP (ms)
	Step 1 (ms)	Step 2 (ms)	Step 3 (ms)	**Total (ms)**	
I	692	2482	21	3195	$\approx 5 \times 10^8$
II	756	3207	9	3972	$\approx 5 \times 10^8$
III	2456	7146	12	9614	$\approx 5 \times 10^8$
IV	4721	16972	9	21702	$\approx 5 \times 10^8$
V	129328	206741	8	336077	$\approx 8 \times 10^8$

as $[\![m_1.\alpha]\!] = [\![m_1]\!]^\alpha$. Hence, a Paillier cryptosystem is an additively homomorphic cryptosystem. Let us denote $E()$ and $D()$ as the Paillier homomorphic encryption and decryption functions. Using the homomorphic properties and the above definitions, homomorphic encryption-based PPSP is described in Table III.

According to NIST recommendation [31], [32], public-key encryption schemes, such as RSA and the Paillier must use 3072-b long keys for encryption and decryption in order to achieve 128-b security. Hence, to obtain the running time for the Paillier homomorphic encryption-based PPSP, we used 3072-b long keys. We also obtained the running time for the proposed LWE-based scheme for the first five sets of parameters given in Table II (the sixth set was ignored as it was taking too much time to run). The running times averaged over 100 executions are listed in Table IV (no parallelization or multithreading was used).

As presented in Table IV, the result of set I has outperformed the other sets. This is due to the fact that even though the security levels are equal across all the sets, when the size for n increases, the matrix $\mathbf{A}$ becomes larger and requires an increased number of multiplications. In turn, this slows down the algorithm. With this observation, we will continue using the parameters that belong to set I for the remainder of our experiments presented in this article. The last column in Table IV shows the average running time for the Paillier scheme. The proposed scheme is at least 10^5 times faster than the Paillier PPSP scheme. The dimensions of the input vectors for these sets are in the range of 20 000–50 000 (see the third column in Table II).

TABLE V
PARAMETERS AND KEY SIZES FOR THE PROPOSED AND PAILLIER-BASED PPSP SCHEMES FOR DIFFERENT LEVELS OF SECURITY

Security	n	m $\approx$	q $\approx$	$\alpha.q$ $\approx$	Paillier Key Size
2^{80}	50	23500	2^{470}	2^{439}	1024
2^{112}	50	27500	2^{550}	2^{518}	2048
2^{128}	50	28500	2^{570}	2^{538}	3072
2^{192}	50	40500	2^{810}	2^{777}	7680
2^{256}	50	50000	2^{1000}	2^{997}	15360

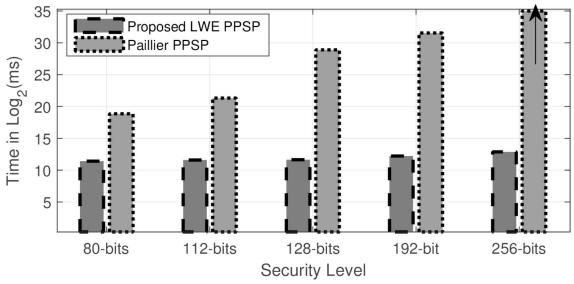

Fig. 2. Average running time for the proposed LWE PPSP scheme against the Paillier PPSP scheme for different security levels. Note that the y-axis is in log scale.

To compare the performance of the proposed scheme for different security levels, a new set of parameters are provided in Table V. Based on the NIST recommendations [31], [32], the key sizes for the Paillier scheme is also provided in Table V. Using this information, the average running time is plotted in Fig. 2. While the average running time for the proposed scheme is increasing linearly, it increases exponentially for the Paillier scheme. It should be noted that the average running time for the proposed scheme is around 8 s at 256-b security (without any parallel computations or multithreading). These results demonstrate that the proposed lattice PPSP scheme is significantly faster than the Paillier PPSP.

B. Proposed Scheme and Randomization Technique

Table VI shows the state-of-the-art randomization-based PPSP [4], [20]. The security of this algorithm depends on the hardness of the factoring an integer, i.e., $C_i = s(a_i.\alpha + c_i) \bmod p$, $a_i \neq 0$. C_is are protected by s and known only to X. If Y wants to recover the X's input vector, Y needs to factor all C_is to find the common s. This approach can be seen as an approach used in RSA encryption or any public-key encryption that relies on the hardness of factoring integers. According to the NIST recommendation [31], [32], the size of these integers must be around 3072 b in order to obtain 128-b security (without loss of generality, we ignore the requirement of prime numbers). Hence, we set $k1$ in Table VI to 3072 b to compare randomization-based PPSP and the proposed lattice PPSP scheme.

TABLE VI
RANDOMIZATION-BASED PPSP ALGORITHM

Input by X: $\mathbf{a} = [a_1, \ldots, a_m]^T \in \{0,1\}^m$ and **Y:** $\mathbf{b} = [b_1, \ldots, b_m]^T \in \{0,1\}^m$
Output to X: $\mathbf{a}^T\mathbf{b}$
Step 1: X performs the following operations:
Given security parameters k_1, k_2, k_3, k_4,
choose two large primes α, p
such that $\lvert p\rvert = k_1$, $\lvert\alpha\rvert = k_2$, set $a_{m+1} = a_{m+2} = 0$
Choose a large random number $s \in Z_p$, and $m+2$ random numbers c_i, $i = 1, 2, ..., m+2$, with $\lvert c_i\rvert = k_3$
FOR EACH a_i, $i = 1, 2, ..., m+2$
Compute
$C_i = s(a_i.\alpha + c_i) \bmod p,\ a_i \neq 0$
$C_i = sc_i \bmod p,\ a_i = 0$
END FOR
keeps $s^{-1} \bmod p$ secret, and sends $(\alpha, p, C_1 \ldots C_{m+2})$ to Y
Step 2: Y executes the following operations
set $b_{m+1} = b_{m+2} = 0$
FOR EACH b_i, $i = 1, 2, ..., m+2$
Compute
$D_i = b_i.\alpha.C_i \bmod p,\ b_i \neq 0$
$D_i = r_i.C_i \bmod p,\ b_i = 0,$
where r_i is a random number with $\lvert r_i\rvert = k_4$
END FOR
Send $D = \sum_{i=1}^{m+2} D_i \bmod p$ to X
Step 3: Now X computes and obtains
$E = s^{-1}.D \bmod p$ and get $\mathbf{a}^T\mathbf{b}$
$= \sum_{i=1}^{n} a_i.b_i = \frac{E-(E \bmod \alpha^2)}{\alpha^2}$.

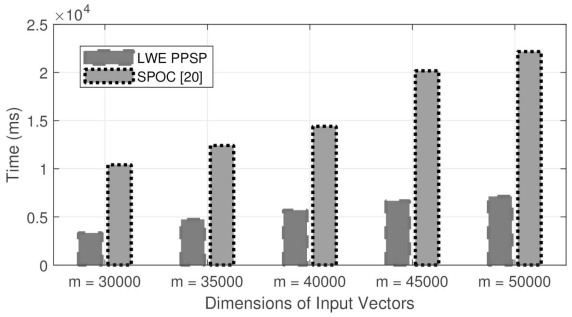

Fig. 3. Average running time for the proposed LWE PPSP scheme against the randomization-based PPSP scheme [4], [20] for different sizes of input vectors.

Using this setting, the average running time for the proposed and randomization-based PPSP schemes are obtained at 128-b security. Fig. 3 shows the average running times for both schemes for different input vectors whose dimensions are between 30 000 and 50 000. The proposed scheme is at least twice as fast compared to the randomization-based scheme for the security parameters. It should be noted that since the randomization-based scheme relies on the hardness of integer factorization, similar to the Paillier scheme, it is also vulnerable for quantum attacks.

Even though the proposed scheme is developed to protect the PP applications against quantum computers, the efficiency analysis shows that the algorithm can be used to replace the existing schemes. Running time in Table IV is obtained from sequential programming. It is taking around 3 s to execute the SP of two vectors whose dimensions are around 30 000. Nearly 2.5 s are spent on step 2 calculating (11). This equation can be computed in parallel, i.e., $\mathbf{t}^T\mathbf{A}$ is equivalent to $\mathbf{t}^T\mathbf{a}_i$ where $i \leq m$. Therefore, we used multithreading features of Java to speedup the process. By setting four threads, the average running time has been reduced to 1.2 from 3 s.

TABLE VII
COMMUNICATION COST COMPARISON

	X to Y	Y to X	Total
Proposed LWE PPSP	3.6 kB	2.1 MB	∼2 MB
Paillier PPSP	11.5 MB	0.3 kB	∼12 MB
Randomisation PPSP	11.5 MB	0.3 kB	∼12 MB

C. Communication Complexity

Using the algorithms in Fig. 1 (the proposed LWE scheme), Table III (the Paillier homomorphic encryption scheme-based PPSP), and Table VI (randomization-based PPSP), we can calculate the communication cost in terms of transmitted bits between entities **X** and **Y**.

1) Total Bits Transmitted From Entity X to Entity Y: Total number of bits required for the proposed LWE-based PPSP scheme is $n * \log_2(q)$. Similarly, $m * \log_2(\text{pub})$ and $(m+4) * \log_2(k1)$ number of bits are required for the Paillier-based scheme and randomization scheme, respectively.

2) Total Bits Transmitted From Entity Y to Entity X: Total number of bits required for the proposed LWE-based PPSP scheme is $(m+1) * \log_2(q)$. Similarly, $\log_2(\text{pub})$ and $\log_2(k1)$ number of bits are required for the Paillier-based scheme and randomization scheme, respectively.

At 128-b level security, if we extract the parameters, then $n = 50$, $\log_2(q) = 570$, $\log_2(\text{pub}) = 3072$, and $\log_2(k1) = 3072$. Using these parameters, Table VII shows the communication cost for all three schemes when the dimension of the input vectors is $m = 30\,000$. It is clear from Table VII that the LWE scheme significantly benefits from a shorter prime number (six times smaller than the other schemes' prime number) and achieves six times lower data requirement to perform the scalar computation.

VII. CONCLUSION

In this article, a novel PPSP computation using the fundamentals of lattice-based cryptography has been proposed. In particular, the proposed scheme was built directly on top of the lattice hard problems, such as the shortest integer solution and LWEs. The 128-b encryption security has been achieved with the proposed framework. Several validation and verification experiments have shown that the proposed scheme is one of the best performing schemes in terms of complexity whilst not compromising systems security.

Challenges and Future Work: The dimensions of the input vectors depend on n and q, i.e., $m = n.\log_2(q)$. Hence, the proposed work supports larger dimensions such as 30 000. Even though, this is appropriate for many applications, converting the solution to support smaller dimensions such as 100 would be an interesting problem that requires further investigations.

ACKNOWLEDGMENT

Source code for this work can be found in Github repo (https://github.com/rahulay1/LWE).

REFERENCES

[1] R. L. Lagendijk, Z. Erkin, and M. Barni, "Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 82–105, Jan. 2013.

[2] M. Barni, P. Failla, R. Lazzeretti, A. R. Sadeghi, and T. Schneider, "Privacy-preserving ECG classification with branching programs and neural networks," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 2, pp. 452–468, Jun. 2011.

[3] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft, "Privacy-preserving face recognition," in *Proc. Int. Symp. Privacy Enhancing Technol. Symp.* Aug. 2009, pp. 235–253.

[4] Y. Rahulamathavan, K. R. Sutharsini, I. G. Ray, R. Lu, and M. Rajarajan, "Privacy-preserving iVector-based speaker verification," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 3, pp. 496–506, Mar. 2019.

[5] Y. Rahulamathavan, R. Phan, S. Veluru, K. Cumanan, and M. Rajarajan, "Privacy-preserving multi-class support vector machine for outsourcing the data classification in cloud," *IEEE Trans. Depend. Secure Comput.*, vol. 11, no. 5, pp. 467–479, Sep./Oct. 2014.

[6] Y. Rahulamathavan, S. Veluru, R. Phan, J. Chambers, and M. Rajarajan, "Privacy-preserving clinical decision support system using Gaussian kernel based classification," *IEEE J. Biomed. Health Inform.*, vol. 18, no. 1, pp. 56–66, Jan. 2014.

[7] Y. Rahulamathavan, R. Phan, J. Chambers, and D. Parish, "Facial expression recognition in the encrypted domain based on local fisher discriminant analysis," *IEEE Trans. Affective Comput.*, vol. 4, no. 1, pp. 83–92, Jan.–Mar. 2013.

[8] Y. Rahulamathavan and M. Rajarajan, "Efficient privacy-preserving facial expression classification," *IEEE Trans. Depend. Secure Comput.*, vol. 14, no. 3, pp. 326–338, Jun. 2017.

[9] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in *Proc. 37th ACM Symp. Theory Comput. (STOC)*, 2005, pp. 84–93.

[10] M. Ajtai, "Generating hard instances of lattice problems," in *Proc. 28th Annu. ACM Symp. Theory Comput.*, Jul. 1996, pp. 99–108.

[11] C. Peikert, "Lattice cryptography for the Internet," in *Proc. Int. Workshop Post Quantum Cryptography*, Oct. 2014, pp. 197–219.

[12] C. Peikert, "A decade of lattice cryptography," *Found. Trends Theor. Comput. Sci.*, vol. 10, no. 4, pp. 283–424, 2016.

[13] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *J. ACM*, vol. 56, no. 6, p. 34, 2009.

[14] N. Gama, and P. Q. Nguyen, "Predicting lattice reduction," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, Apr. 2008, pp. 31–51.

[15] A. Blum, A. Kalai, and H. Wasserman, "Noise-tolerant learning, the parity problem, and the statistical query model," *J. ACM*, vol. 50, no. 4, pp. 506–519, 2003.

[16] D. Micciancio, and C. Peikert, "Trapdoors for lattices: Simpler, tighter, faster, smaller," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, Apr. 2012, pp. 700–718.

[17] D. Micciancio and O. Regev, "Lattice-based cryptography," in *Post-Quantum Cryptography*. Berlin, Germany: Springer, 2009, pp. 147–191.

[18] S. Agrawal, D. M. Freeman, and V. Vaikuntanathan, "Functional encryption for inner product predicates from learning with errors," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Security*, Dec. 2011, pp. 21–40.

[19] C. Meyer, *Matrix Analysis and Applied Linear Algebra*. Philadelphia, PA, USA: Soc. Ind. Appl. Math., 2000.

[20] R. Lu, H. Zhu, X. Liu, J. Liu, and J. Shao, "Toward efficient and privacy-preserving computing in big data era," *IEEE Netw.*, vol. 28, no. 4, pp. 46–50, Jul./Aug. 2014.

[21] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. 17th Int. Conf. Theory Appl. Cryptograph. Techn. (EUROCRYPT'99)*, 1999, pp. 223–238.

[22] Q. Wang, S. Hu, K. Ren, M. He, M. Du, and Z. Wang, "CloudBI: Practical privacy-preserving outsourcing of biometric identification in the cloud," in *Proc. Eur. Symp. Res. Comput. Security*, Sep. 2015, pp. 186–205.

[23] S. Hu, M. Li, Q. Wang, S. S. Chow, and M. Du, "Outsourced biometric identification with privacy," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 10, pp. 2448–2463, Oct. 2018.

[24] W. Du and M. J. Atallah, "Privacy-preserving cooperative statistical analysis," in *Proc. IEEE 17th Annu. Comput. Security Appl. Conf. (ACSAC)*, New Orleans, LA, USA, Dec. 2001, pp. 102–110.

[25] W. Du and Z. Zhan, "Building decision tree classifier on private data," in *Proc. IEEE Int. Conf. Privacy Security Data Min.*, vol. 14, Dec. 2002, pp. 1–8.

[26] J. Vaidya and C. Clifton, "Privacy preserving association rule mining in vertically partitioned data," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discover. Data Min.*, Jul. 2002, pp. 639–644.

[27] A. Amirbekyan and V. Estivill-Castro, "A new efficient privacy-preserving scalar product protocol," in *Proc. 6th Aust. Conf. Data Min. Anal.*, vol. 70, Dec. 2007, pp. 209–214.

[28] R. Zhang, Y. Zhang, J. Sun, and G. Yan, "Fine-grained private matching for proximity-based mobile social networking," in *Proc. IEEE INFOCOM*, Orlando, FL, USA, Mar. 2012, pp. 1969–1977.

[29] W. Dong, V. Dave, L. Qiu, and Y. Zhang, "Secure friend discovery in mobile social networks," in *Proc. IEEE INFOCOM*, Shanghai, China, Apr. 2011, pp. 1647–1655.

[30] B. Goethals, S. Laur, H. Lipmaa, and T. Mielikäinen, "On private scalar product computation for privacy-preserving data mining," in *Proc. Int. Conf. Inf. Security Cryptol.*, Dec. 2004, pp. 104–120.

[31] E. Barker and A. Roginsky. (Nov. 2015). *Transitions: Recommendation for Transitioning The Use of Cryptographic Algorithms and Key Lengths, NIST SP-800-131A Rev 1*. [Online]. Available: Nvlpubs.nist.gov

[32] (Jan. 2016). *NIST Special Publication 800-57 Part 1 Revision 4: Recommendation for Key Management*. [Online]. Available: http://csrc.nist.gov/publications/PubsSPs.html800-57pt1r4

[33] M. R. Albrecht, R. Player, and S. Scott, "On the concrete hardness of learning with errors," *J. Math. Cryptol.*, vol. 9, no. 3, pp. 169–203, Oct. 2015, doi: 10.1515/jmc-2015-0016.

[34] J. H. Cheon, A. Kim, and D. Yhee, "Multi-dimensional packing for HEAAN for approximate matrix arithmetics," Cryptol. ePrint Archive, Lyon, France, Rep. 2018/1245, 2018.

[35] T. Graepel, K. Lauter, and M. Naehrig, "ML confidential: Machine learning on encrypted data," in *Proc. Int. Conf. Inf. Security Cryptol.*, Nov. 2012, pp. 1–21.

[36] J. W. Bos, W. Castryck, I. Iliashenko, and F. Vercauteren, "Privacy-friendly forecasting for the smart grid using homomorphic encryption and the group method of data handling," in *Proc. Int. Conf. Cryptol. Africa*, May 2017, pp. 184–201.

[37] J. L. Crawford, C. Gentry, S. Halevi, D. Platt, and V. Shoup, "Doing real work with FHE: The case of logistic regression," in *Proc. 6th Workshop Encrypted Comput. Appl. Homomorphic Cryptography*, Oct. 2018, pp. 1–12.

Yogachandran Rahulamathavan (Member, IEEE) received the B.Sc. degree (First-Class Hons.) in electronic and telecommunication engineering from the University of Moratuwa, Moratuwa, Sri Lanka, in 2008, and the Ph.D. degree in signal processing from Loughborough University, Loughborough, U.K., in 2011.

He is a Senior Lecturer and the Program Director of the M.Sc. Cyber Security and Big Data Program with Loughborough University London, London, U.K. He is currently coordinating U.K.–India project between Loughborough, IIT Kharagpur and City, University of London. His research interest is on developing novel security protocols to advance machine learning techniques to solve complex privacy issues in emerging applications, e.g., patient's healthcare data sharing, biometric authentication systems, and identity management in cloud.

Dr. Rahulamathavan is an Associate Editor of IEEE ACCESS.

Safak Dogan (Senior Member, IEEE) received the B.Sc. degree from the Istanbul Technical University, Istanbul, Turkey, in 1995, and the M.Sc. and Ph.D. degrees from the University of Surrey, Guildford, U.K., in 1996 and 2001, respectively.

He is a Senior Lecturer of multimedia technologies with the Institute for Digital Technologies, Loughborough University London, London, U.K. He has managed various EU-funded multinational collaborative research projects. His main areas of expertise include digital media signal processing, multimedia communication systems and networks, and quality assessment. His recent research focuses on data visualization and user activity analysis for privacy protection.

Xiyu Shi (Member, IEEE) received the B.Eng. degree in radio engineering from Southeast University, Nanjing, China, in 1984, the M.Sc. degree in communication and electronic systems from Beijing University of Aeronautics and Astronautics, Beijing, China, in 1989, and the Ph.D. degree in computer networks from Cranfield University, Shrivenham, U.K., in 2002.

From 2002 to 2004, he was with the Centre for Communication Systems Research, University of Surrey, Guildford, U.K. From 2004 to 2005, he was with the Department of Applied Computing, University of Buckingham, Buckingham, U.K. In 2005, he rejoined the University of Surrey as a Research Fellow. He is currently a Lecturer with the Institute for Digital Technologies, Loughborough University London, London, U.K. His research interests include IoT-related privacy and security issues, cybersecurity, audio signal processing, and multimedia communications.

Rongxing Lu (Senior Member, IEEE) received the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, in 2012.

He worked as an Assistant Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, from April 2013 to August 2016. He is currently an Associate Professor with the Faculty of Computer Science (FCS), University of New Brunswick (UNB), Fredericton, NB, Canada. He worked as a Postdoctoral Fellow with the University of Waterloo from May 2012 to April 2013. His research interests include applied cryptography, privacy enhancing technologies, and IoT-big data security and privacy. He has published extensively in his areas of expertise.

Dr. Lu was awarded the most prestigious "Governor General's Gold Medal" and won the 8th IEEE Communications Society (ComSoc) Asia–Pacific Outstanding Young Researcher Award in 2013. He was the recipient of eight best paper awards from some reputable journals and conferences. He was the Winner of 2016–2017 Excellence in Teaching Award, FCS, UNB. He is currently a Senior Member of the IEEE Communications Society. He currently serves as the Vice-Chair (Conferences) for IEEE ComSoc Communications and Information Security Technical Committee.

Muttukrishnan Rajarajan (Senior Member, IEEE) received the B.Eng. and Ph.D. degrees from City University London, London, U.K., in 1994 and 1999, respectively.

In 1999, he worked with City University London as a Research Fellow. In August 2000, he moved to Logica as a Telecommunication Consultant. After a few years in the industry, he is currently a Professor of security engineering. He is also the Programme Director for the Engineering with Management and Entrepreneurship Programme.

Prof. Rajarajan also sits on the Editorial boards of *Wireless Networks* (Springer/ACM), *Health Policy and Technology* (Elsevier), and *Information Management and Computer Security* (Emerald). He is a member of IET and an Associate Member of the Institute of Information Security Professionals and a member of Technical Programme Committees for various prestigious conferences.

Ahmet Kondoz (Senior Member, IEEE) received the Ph.D. degree from the University of Surrey, Guildford, U.K., in 1987.

He was a Research Fellow with the Communication Systems Research Group, University of Surrey from 1986 to 1988 and became a Lecturer in 1988, a Reader in 1995, and in 1996, he was promoted to a Professor of multimedia communication systems. He was the Founding Head of I-LAB, a multidisciplinary multimedia communication systems research lab with the University of Surrey. Since 2014, he has been the Founding Director of the Institute for Digital Technologies, Loughborough University London, London, U.K., a post graduate teaching, research, and enterprise institute. He has over 400 publications, including six books, several book chapters, and seven patents, and graduated over 75 Ph.D. students. He has been a Consultant for major wireless media industries and has been acting as an Advisor for various international governmental departments, research councils, and patent attorneys. His research interests include digital signal processing and coding, fixed and mobile multimedia communication systems, 3-D immersive media applications for the future Internet systems, smart systems, such as autonomous vehicles and assistive technologies, big data analytics, and visualization and related cybersecurity systems.

Dr. Kondoz has been involved with several European Commission FP6 & FP7 research and development projects, such as NEWCOM, e-SENSE, SUIT, VISNET, and MUSCADE. Involving leading universities, research institutes, and industrial organizations across Europe. He coordinated FP6 VISNET II NoE, FP7 DIOMEDES STREP, and ROMEO IP projects, involving many leading organizations across Europe which deals with the hybrid delivery of high-quality 3-D immersive media to remote collaborating users including those with mobile terminals. He co-chaired the European networked media advisory task force, and contributed to the Future Media and 3-D Internet activities to support the European Commission in the FP7 programmes.