IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 18, 2019, accepted May 2, 2019, date of publication May 7, 2019, date of current version May 20, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2915196

Efficient and Secure Outsourcing of DFT, IDFT, and Circular Convolution

XIANGLI XIAO[1], JUNJIAN HUANG[2], YUSHU ZHANG[3,4], (Member, IEEE), AND XING HE[1]

[1]School of Electronics and Information Engineering, Southwest University, Chongqing 400715, China
[2]Key Laboratory of Machine Perception and Children's Intelligence Development, Chongqing University of Education, Chongqing 400067, China
[3]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
[4]Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China

Corresponding authors: Junjian Huang (hmomu@sina.com) and Xing He (hexingdoc@hotmail.com)

This work was supported in part by the Guangxi Key Laboratory of Trusted Software under Grant KX201904, in part by the Fundamental Research Funds for the Central Universities under Grant XDJK2019B010, in part by the Foundation of Key Laboratory of Machine Perception and Children's Intelligence Development under Grant 16xjpt07, in part by the National Natural Science Foundation of China under Grant 61462032 and Grant 61773320, in part by the Natural Science Foundation Project of Chongqing CSTC under Grant cstc2018jcyjAX0583 and Grant cstc2018jcyjAX0810, and in part by the Foundation of Chongqing University of Education under Grant 18GZKP02.

ABSTRACT Discrete Fourier transform (DFT), inverse DFT (IDFT), and circular convolution are important tools for analyzing and designing discrete signals and systems, and are widely used in various industries. In order to pursue faster operational efficiencies or more accurate operational results, engineering calculations are often required to be quick and easy. Therefore, it is necessary to reduce the local computational overhead required to perform DFT, IDFT, and circular convolution. In addition to improving the algorithms themselves, cloud computing outsourcing is also an optional method. In this paper, in response to the new challenges brought by cloud computing outsourcing, we design and propose an efficient and secure cloud computing outsourcing protocol for DFT, IDFT, and circular convolution. Through a theoretical explanation and simulation experiment, we show the efficiency, security, and verifiability of the proposed protocol.

INDEX TERMS Computing outsourcing, discrete Fourier transform, circular convolution, cloud computing.

I. INTRODUCTION

In the field of signal processing, spectral analysis of signals and systems is often required. Different from the Fourier analysis of continuous signals and systems, DFT is discretized in both time domain and frequency domain, which is suitable for numerical operation and becomes a powerful tool for computer to analyze the spectrum of signals and systems. Therefore, DFT is widely used in digital communication [1], speech signal processing [2], image processing [3], video coding [4], power spectrum estimation [5], system analysis and simulation [6], radar signal processing [7], seismic analysis [8]. Meanwhile, IDFT and cyclic convolution, as companion algorithms of DFT, are usually used together in the spectral analysis.

In practical engineering applications, it is often necessary to perform real-time processing on the acquired signals, which requires the signal processing equipment to operate at a fast enough speed. Especially with the rapid development of high-speed mobile network [9], [10] and Internet of Things [11], [12], higher speed requirements are imposed on real-time signal processing. However, the direct calculation of DFT, IDFT, and circular convolution is proportional to the square of the transform interval length N, which is too large. When the output precision is required to be high, the number of sampling points of the signal must be increased, i.e., N will become quite large. At this point, it becomes impractical to use simple devices to process signals in real time. Hence, in 1965, Cooley *et al.* proposed a fast algorithm for DFT in [13]. Since then, the improved algorithms have been continuously proposed, forming a set of efficient computing methods, which is now the fast Fourier transform (FFT). The emergence of FFT greatly improves the computational efficiency of DFT, which creates conditions for digital signal processing technology to be applied to real-time processing of various signals and enables the simplification of equipment.

In this paper, we will address the problem from another perspective, namely, the use of cloud computing outsourcing to improve local computing efficiency. In the cloud

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

computing platform, there is a huge data computing and management system with powerful computing power and various application software, which provides service to clients in a leased manner. With cloud computing, clients can offload computation-intensive computing to the cloud, thereby saving computing overhead or performing locally unexecutable operations.

However, the use of cloud computing raises a number of new challenges. Firstly, since the cloud is untrustworthy, we need to find ways to fully protect the privacy of our clients from being stolen by the cloud, i.e., encrypting the input and output information. Secondly, to save computing resources or costs, the cloud may return a random error result and hope that it will not be discovered by the client. Even if the cloud is honest, the results returned may still be wrong due to possible software bugs or hardware errors. Thus, it is essential that the client performs efficient results verification locally. Thirdly, clients must be able to obtain considerable savings in computing overhead from outsourcing; otherwise, they will have no need to perform any computing outsourcing. That is to say, the computational cost required by the client to perform encryption, decryption, and verification operations must be much less than that required for direct local calculations. In general, cloud computing outsourcing protocols must be privacy-preserving, verifiable, and efficient.

To meet the above three challenges, we first conduct efficient encryption for the original problem. Hence, the cloud cannot obtain any private information from the input and output. Afterwards, Parseval's theorem and mathematical definitions of DFT and IDFT are used to efficiently verify the results returned from the cloud, and the client can detect the error result with a probability of nearly 1. Finally, since the encryption, decryption, and verification algorithms used are all quite efficient, the client can obtain huge computational savings by outsourcing. Theoretical explanation and simulation results show that as long as N is not too small, the computational cost savings achieved by outsourcing are more prominent than those achieved by FFT.

We summarize our key contributions as follows.

- We propose an efficient and secure cloud computing outsourcing protocol for DFT, IDFT, and circular convolution.
- Theoretical explanation and simulation experiment show that the proposed protocol can achieve considerable overhead savings for the client.
- The proposed protocol not only fully protects the privacy information of the client, but also can detect the misbehavior of the cloud with a probability of nearly 1.
- The proposed protocol not only has no restrictions on the value of N, but also flexibly adjusts the trade-offs between security, efficiency, and accuracy.

The rest of this paper is organized as follows. Section II introduces some essential preliminaries. In Section III, we describe our proposed outsourcing protocol in detail. Then, we provide a theoretical analysis of the protocol performance in Section IV. Section V presents our experiment results and Section VI overviews the related work. Finally, the last section gives concluding remarks.

II. PROBLEM FORMULATION

In this section, we first present the definitions of DFT, IDFT, and circular convolution, and then the system model, threat model, and design goals are given.

A. DEFINITIONS OF DFT, IDFT, AND CIRCULAR CONVOLUTION

1) DFT AND IDFT

Let $x(n)$ be a finite-length sequence of length M, then define the N-point DFT of $x(n)$ as

$$\begin{aligned} X(k) &= DFT[x(n)]_N \\ &= \sum_{n=0}^{N-1} x(n) W_N^{kn}, \quad k = 0, 1, \ldots, N-1, \end{aligned} \tag{1}$$

where $W_N = exp(-j\frac{2\pi}{N})$, N is called the DFT transform interval length, and $N \geq M$. Afterwards, the IDFT of $X(k)$ is defined as

$$\begin{aligned} x(n) &= IDFT[X(k)]_N \\ &= \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-kn}, \quad n = 0, 1, \ldots, N-1. \end{aligned} \tag{2}$$

Generally, (1) and (2) are referred to as discrete Fourier transform pairs and abbreviated as $DFT[x(n)]_N$ and $IDFT[X(k)]_N$, respectively.

2) CIRCULAR CONVOLUTION

Let $x_1(n)$ and $x_2(n)$ be two finite-length sequences of length N_1 and N_2, respectively, then define the L-point circular convolution of $x_1(n)$ and $x_2(n)$ as

$$\begin{aligned} y_c(n) &= x_2(n) \textcircled{L} x_1(n) \\ &= [\sum_{m=0}^{L-1} x_2(m) x_1((n-m))_L] R_L(n), \end{aligned} \tag{3}$$

where $L \geq max[N_1, N_2]$, L is called the circular convolution interval length, $R_L(n)$ is a rectangular sequence, and $x_1((n-m))_L$ can be regarded as the result of periodic continuation after cyclic shift of $x_1(n)$ [14].

B. SYSTEM MODEL, THREAT MODEL, AND DESIGN GOALS

1) SYSTEM MODEL

As shown in Fig. 1, a client wants to outsource a high-complexity original problem to the cloud to reduce local computing overhead. First of all, since the client does not want the cloud to obtain any relevant private information from the input and output, the client uses a private key to efficiently encrypt the original problem locally, resulting in the encrypted problem. Afterwards, the client sends the encrypted problem to the cloud. Once receiving the encrypted problem, the cloud executes the corresponding solution algorithm to solve the encrypted problem. The cloud

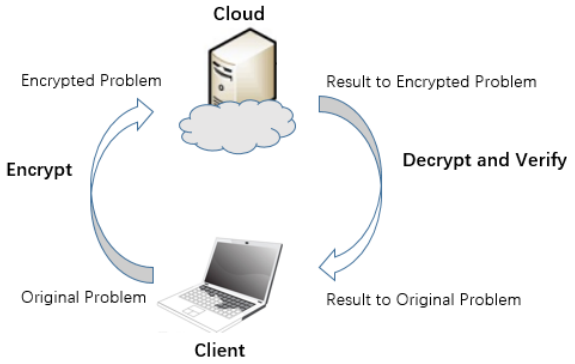


FIGURE 1. Secure outsourcing system model.

then returns the result of the encrypted problem to the client. After receiving the result, the client uses the private key to decrypt the returned result locally and obtains the result to the original problem. Finally, the client performs an efficient result validation algorithm locally to verify the correctness of the result. If it is correct, the client receives the result; otherwise, it indicates a computational error or fraud in the cloud and the client rejects the result.

2) THREAT MODEL AND DESIGN GOALS

Generally, the threat model faced by the cloud computing outsourcing system is divided into two levels: semi-honest cloud model and malicious cloud model [15]. As described earlier, we assume that the cloud is malicious in our protocol. In the malicious cloud model, on the one hand, the cloud attempts to obtain the privacy of the client from its only known input and output information; on the other hand, the cloud may deliberately return a random error result to save costs and hopes that it will not be detected by the client. For such a malicious cloud model, we formulate the design goals as follows.

- **Correctness**: If the cloud and the client follow the protocol carefully, the client must be able to obtain the correct answer.
- **Privacy**: The cloud cannot steal any private information from the protected input and output.
- **Soundness**: The client can detect the error result returned by the cloud with a probability of nearly 1.
- **Efficiency**: The client must be able to obtain considerable computational overhead savings from outsourcing.

III. PROTOCOL CONSTRUCTION

In this section, we will construct the outsourcing protocols for DFT, IDFT, and circular convolution, respectively.

A. EFFICIENT AND SECURE OUTSOURCING OF DFT

To protect the privacy sequence $x(n)$ of length M, the client generates two random real numbers a, b (for better security, a and b can also be complex numbers) and a random integer m $(0 < m < M)$, and perform

$$y(n) = a{\cdot}x(n) + b{\cdot}x((n+m))_N R_N(n), \tag{4}$$

where $x((n+m))_N R_N(n)$ is the cyclic shift of $x(n)$ and $N \geq M$. Then the client sends $y(n)$ to the cloud, and the cloud solves $DFT[y(n)]_N$ after receiving $y(n)$.

According to the time domain cyclic shift theorem and linear property of DFT [14], we have

$$DFT[x((n+m))_N R_N(n)]_N = W_N^{-km} X(k), \tag{5}$$

$$DFT[y(n)]_N = a{\cdot}DFT[x(n)]_N + b{\cdot}DFT[x((n+m))_N R_N(n)]_N, \tag{6}$$

where $X(k) = DFT[x(n)]_N$.

Consequently, we can obtain that

$$Y(k) = DFT[y(n)]_N = a{\cdot}X(k) + b{\cdot}W_N^{-km} X(k), \tag{7}$$

where $k = 0, 1, \ldots, N-1$. Afterwards, the cloud returns $Y(k)$ to the client. Deformation of (7) gives

$$X(k) = DFT[x(n)]_N = \frac{Y(k)}{a + b{\cdot}W_N^{-km}}, \tag{8}$$

where $k = 0, 1, \ldots, N-1$. Thus, the client can obtain the required $X(k)$ by performing (8) to efficiently decrypt the returned $Y(k)$.

In the case of $n = 0$, (2) gives

$$N{\cdot}x(0) = \sum_{k=0}^{N-1} X(k). \tag{9}$$

Moreover, according to the Parseval's theorem, we have

$$N{\cdot}\sum_{n=0}^{N-1} |x(n)|^2 = \sum_{k=0}^{N-1} |X(k)|^2, \tag{10}$$

where $|x(n)|$ and $|X(k)|$ represent the size of $x(n)$ and $X(k)$, respectively. Therefore, the client can verify the correctness of $X(k)$ by (9) and (10). If (9) and (10) are true within the error tolerance, then the result $X(k)$ is correct; otherwise, it is wrong.

Algorithm 1 Outsourcing Protocol for DFT

1: Initialization: Input a finite-length sequence $x(n)$ of length M, the client generates two random real numbers a, b and a random integer m $(0 < m < M)$;
2: The client encrypts $x(n)$ by (4) to obtain $y(n)$;
3: The client sends $y(n)$ to the cloud;
4: The cloud solves $DFT[y(n)]_N$ to obtain $Y(k)$;
5: The cloud sends the result $Y(k)$ to the client;
6: The client decrypts $Y(k)$ by (8) to obtain the required result $X(k)$;
7: The client verifies $X(k)$ by (9) and (10). If $X(k)$ passes the check, accept it; otherwise, reject it;
8: End.

We summarize the proposed outsourcing protocol for DFT in Algorithm 1.

B. EFFICIENT AND SECURE OUTSOURCING OF IDFT

Similarly, to protect the private $X(k)$, the client generates two random real numbers c, d (for better security, c and d can also be complex numbers) and a random integer l $(0 < l < N)$, and perform

$$Y(k) = c \cdot X(k) + d \cdot X((k+l))_N R_N(k). \tag{11}$$

Then the client sends $Y(k)$ to the cloud, and the cloud solves $IDFT[Y(k)]_N$ after receiving $Y(k)$. According to the frequency domain cyclic shift theorem and linear property of DFT [14], we have

$$IDFT[X((k+l))_N R_N(k)]_N = W_N^{nl} x(n), \tag{12}$$

$$IDFT[Y(k)]_N = c \cdot IDFT[X(k)]_N + d \cdot IDFT[X((k+l))_N R_N(k)]_N, \tag{13}$$

where $x(n) = IDFT[X(k)]_N$.

Consequently, we can obtain that

$$y(n) = IDFT[Y(k)]_N = c \cdot x(n) + d \cdot W_N^{nl} x(n), \tag{14}$$

where $n = 0, 1, \ldots, N-1$. Afterwards, the cloud returns $y(n)$ to the client. Deformation of (14) gives

$$x(n) = IDFT[X(k)]_N = \frac{y(n)}{c + d \cdot W_N^{nl}}, \tag{15}$$

where $n = 0, 1, \ldots, N-1$. Thus, The client can obtain the required $x(n)$ by performing (15) to efficiently decrypt the returned $y(n)$.

In the case of $k = 0$, (1) gives

$$X(0) = \sum_{n=0}^{N-1} x(n). \tag{16}$$

Therefore, the client can verify the correctness of $x(n)$ by (10) and (16).

Algorithm 2 Outsourcing Protocol for IDFT

1: Initialization: Input $X(k)$, the client generates two random real numbers c, d and a random integer l $(0 < l < N)$;
2: The client encrypts $X(k)$ by (11) to obtain $Y(k)$;
3: The client sends $Y(k)$ to the cloud;
4: The cloud solves $IDFT[Y(k)]_N$ to obtain $y(n)$;
5: The cloud sends the result $y(n)$ to the client;
6: The client decrypts $y(n)$ by (15) to obtain the required result $x(n)$;
7: The client verifies $x(n)$ by (10) and (16). If $x(n)$ passes the check, accept it; otherwise, reject it;
8: End.

We summarize the proposed outsourcing protocol for IDFT in Algorithm 2.

C. EFFICIENT AND SECURE OUTSOURCING OF CIRCULAR CONVOLUTION

According to the time domain circular convolution theorem [14], we derive the L-point DFT of $y_c(n)$ as

$$Y_c(k) = DFT[y_c(n)]_L = X_1(k)X_2(k), \quad k = 0, 1, \ldots, N-1, \tag{17}$$

where

$$X_1(k) = DFT[x_1(n)]_L, \quad X_2(k) = DFT[x_2(n)]_L. \tag{18}$$

Performing IDFT on (17) yields

$$y_c(n) = IDFT[Y_c(k)]_L. \tag{19}$$

Thus, we can arrive an outsourcing protocol for circular convolution as shown in Algorithm 3.

Algorithm 3 Outsourcing Protocol for IDFT

1: Initialization: Input two finite-length sequences $x_1(n)$ and $x_2(n)$ of length N_1 and N_2, respectively;
2: The client outsources $DFT[x_1(n)]_L$ and $DFT[x_2(n)]_L$ to the cloud by Algorithm 1, and obtains the results $X_1(k)$ and $X_2(k)$;
3: The client performs (17) and obtains the result $Y_c(k)$;
4: The client outsources $IDFT[Y_c(k)]_L$ to the cloud by Algorithm 2, and obtains the result $y_c(n)$;
5: End.

IV. PERFORMANCE EVALUATION

In this section, we present the theoretical analysis of the proposed outsourcing protocol in terms of privacy, soundness, and efficiency.

A. PRIVACY ANALYSIS

For the outsourcing of DFT, the client's input privacy data is $x(n)$, and the output privacy data is $X(k)$, which are encrypted by (4) and (7) respectively. In (4) and (7), since the complex numbers a and b and the integer m are private keys randomly generated by the client locally, the cloud knows nothing about them. Considering that a, b, and m can form an extremely large key space, it is impossible for the cloud to perform a brute-force attack to crack $x(n)$ or $X(k)$ in polynomial time. Therefore, it would be impossible for the cloud to recover $x(n)$ and $y(n)$ from $y(n)$ and $Y(k)$ by trivial means. Moreover, similar to the one-time pad, each group a, b, and m will be replaced by the client after being used once, which further enhances the security of our protocol. In this way, both input privacy and output privacy are protected during the outsourcing of the DFT. Similarly, we can conclude that input privacy and output privacy are also well protected in the outsourcing of IDFT. For the outsourcing of circular convolution, it can be seen as a simple superposition of multi-round DFT outsourcing and IDFT outsourcing, so the client's input and output privacy information is still safe.

TABLE 1. The equivalent conversions of computational complexity for different types of calculations.

Calculation types	Equivalent conversions I	Equivalent conversions II
Multiplication of two complex numbers	4 multiplications of two real numbers + 2 additions of two real numbers	4.2 multiplications of two real numbers
Division of two complex numbers	6 multiplications of two real numbers + 2 divisions of two real numbers + 3 additions of two real numbers	10.3 multiplications of two real numbers
Addition of two complex numbers	2 additions of two real numbers	0.2 multiplications of two real numbers
Multiplication of a complex number and a real number	2 multiplications of two real numbers	2 multiplications of two real numbers
Addition of a complex number and a real number	1 addition of two real numbers	0.1 multiplications of two real numbers

Finally, note that there is a trade-off between security and efficiency in our protocol. If the client has higher security requirements, a, b, c, and d can be set as complex numbers; otherwise, real numbers.

B. VERIFICATION ANALYSIS

In the proposed protocol, the client verifies the results returned from the cloud based on the Parseval's theorem and the mathematical definitions of DFT and IDFT shown in (10), (16), and (9), respectively. Therefore, if each result returned by the cloud is verified, it is easy to know that the client can detect the wrong result with a probability of nearly 1. However, considering that in order to pursue more significant local computing overhead savings, it is also feasible to verify only the parts of the returned results in actual use. The trade-off between efficiency and accuracy can be determined based on different application scenarios.

C. EFFICIENCY ANALYSIS

In this subsection, we first discuss the equivalent conversions of computational complexity for different types of calculations, and then analyze the efficiency of the proposed protocol from the perspective of client-side overhead, cloud-side overhead and communication overhead, and compare it with the case where the client directly uses FFT algorithm. The following analysis takes a, b, c, and d as real numbers and verifies each returned result.

1) EQUIVALENT CONVERSION DISCUSSION

(20) shows the multiplication of two complex numbers:

$$(A + Bi)(C + Di) = (AC - BD) + (AD + BC)i. \quad (20)$$

Therefore, we can obtain that 1 multiplication of two complex numbers can be transformed into 4 multiplications of two real numbers and 2 additions of two real numbers without changing the computational complexity.

Generally, when the operation data is 8 bits, the computer is about 10 times slower to calculate the real number multiplication than the real number addition, and about 20 times slower to calculate the real number division than the real number addition. Based on this relation, we further simplify and obtain that 1 multiplication of two complex numbers can be transformed into 4.2 multiplications of two real numbers without changing the computational complexity.

Through similar derivations, we obtain the conversion relationships shown in Table 1.

2) CLIENT-SIDE OVERHEAD

For the outsourcing of DFT, the computational overhead of the client is concentrated in (4), (8), (9), and (10). Among them, (4) is used to protect the input privacy sequence $x(n)$ of the client, (8) is used to decrypt the result $Y(k)$ returned from the cloud, and (9) and (10) are used to verify the correctness of the result $X(k)$. If $x(n)$ is a complex sequence, then (4) requires $2N$ multiplications of a complex number and a real number and N additions of two complex numbers, (8) requires N multiplications of a complex number and a real number, N divisions of two complex numbers and N additions of a complex number and a real number, (9) requires $4N + 1$ multiplications of two real numbers and $4N - 2$ additions of two real numbers, and (10) requires $N - 1$ additions of two complex numbers and one multiplication of a complex number and a real number. Assuming that the client does not use outsourcing, directly using the definition to solve DFT requires N^2 multiplications of two complex numbers and N^2 additions of two complex numbers, or using the most classical radix-2 FFT algorithm in FFT, then $\frac{N \log_2^N}{2}$ multiplications of two complex numbers and $N \log_2^N$ additions of two complex numbers are required. Finally, we summarize the client-side overhead as shown in Table 2.

We visualized the equivalent conversion results of radix-2 FFT and computing outsourcing in Table 2 to form Fig. 2. From Fig. 2, we can clearly see that when $N \geq 596$, using computing outsourcing can achieve more client-side overhead savings than using radix-2 FFT. Moreover, the greater the N, the more obvious the efficiency advantage of outsourcing. Although the radix-2 FFT algorithm is not the most efficient in FFT, for example, the split-radix FFT algorithm can reduce the calculation overhead by nearly one-third compared with the radix-2 FFT algorithm. However, our outsourcing protocol can further increase overhead savings by adjusting the trade-offs between efficiency, security, and accuracy. In addition, it should be noted that the advantage of outsourcing is also reflected in the value of N is not limited, but FFT usually has a limit on the value of N, for example, radix-2 FFT requires that N must be the power of two.

For IDFT, the client-side overhead of outsourcing or using IFFT (inverse fast Fourier transform) is almost the same as

TABLE 2. Client-side overhead in different scenarios.

Calculation types	DFT	Radix-2 FFT	Computing outsourcing
Multiplication of two complex numbers	N^2	$\frac{N\log_2^N}{2}$	0
Division of two complex numbers	0	0	N
Addition of two complex numbers	N^2	$N\log_2^N$	$2N-1$
Multiplication of a complex number and a real number	0	0	$3N+1$
Addition of a complex number and a real number	0	0	N
Multiplication of two real numbers	0	0	$4N+1$
Addition of two real numbers	0	0	$4N-2$
Equivalent conversions	$4.4N^2$	$2.3N\log_2^N$	$21.2N+2.6$

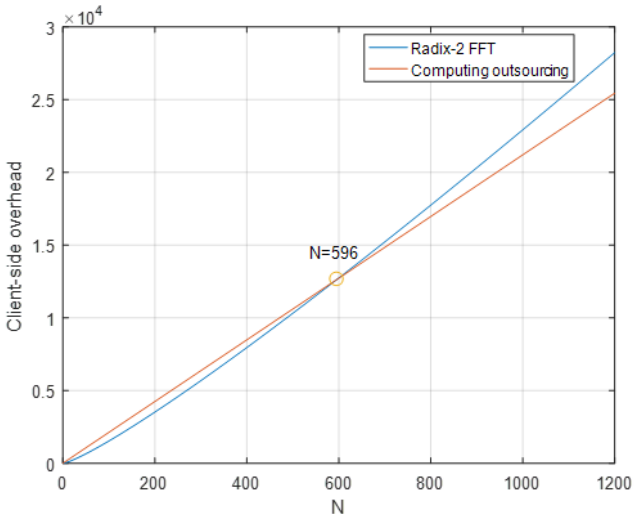


FIGURE 2. Efficiency comparison between radix-2 FFT and computing outsourcing.

for DFT. For circular convolution, the client-side overhead of outsourcing or using (I)FFT is about three times that of DFT. When $x(n)$ is a real sequence, both the outsourcing method and the FFT method can reduce the client-side overhead equally by utilizing the conjugate symmetry of the DFT [14]. Due to space limitations, the detailed discussion is omitted here.

3) CLOUD-SIDE OVERHEAD

In the proposed protocol, the encrypted problem that the client sends to the cloud is still a DFT problem or an IDFT problem. Therefore, the cloud can directly use FFT (or IFFT) to solve them. In other words, our protocol will not bring additional computing overhead to the cloud. Due to the powerful computing power, the speed of solving DFT in the cloud is particularly fast, so it will not drag down the running speed of the client's local device.

4) COMMUNICATION OVERHEAD

For the outsourcing of DFT, the client only needs to send the encrypted sequence $y(n)$ and receive the result $Y(k)$, that is, the communication overhead is only two vectors. For the outsourcing of IDFT, the communication overhead is the same as outsourcing DFT. For the outsourcing of circular convolution, the communication overhead is only six vectors, i.e., the vectors formed after the encryption of $x_1(n)$, $x_2(n)$, $X_1(k)$, $X_2(k)$, $y_c(n)$ and $Y_c(k)$. Therefore, we can conclude that the communication overhead is particularly small in the proposed protocol.

V. EXPERIMENT RESULTS

In this section, we implement our proposed protocol in a numerical experiment to evaluate its performance. The experiment is performed using Matlab 2016b on a PC with an Intel Core i5 processor and 8 GB RAM simulating a client. We separately simulate the client's use of the definition method, the radix-2 FFT method and the outsourcing method to solve DFT, and compare their execution time on the client side. To measure efficiency, we define the following three parameters:

- t_{DFT} represents the time taken by the client to solve DFT using the definition method.
- $t_{radix-2\ FFT}$ represents the time taken by the client to solve DFT using the radix-2 FFT method.
- $t_{outsourcing}$ represents the time taken by the client to solve DFT using the outsourcing method.

Hence, $\frac{t_{DFT}}{t_{outsourcing}}$ represents the performance gain compared to the client using the definition method, and $\frac{t_{radix-2\ FFT}}{t_{outsourcing}}$ represents the performance gain compared to the client using the radix-2 FFT method. In the experiment, we also set a, b, c, and d as real numbers, and verify each returned result. For different values of N, the experimental results are shown in Table 3.

As can be seen from Table 3, the outsourcing method can significantly reduce the local computing overhead compared with the definition method. Compared with the radix-2 FFT method, when N is not too small, the outsourcing method also has an advantage in terms of efficiency. Moreover, the larger N is, the more obvious this advantage will be. For example, the client can obtain an overhead savings of more than $\times 1.50$ when $N \geq 10192$. This is consistent with the results of previous theoretical analysis, and also reflects the practicality of our outsourcing protocol.

VI. RELATED WORK

Cloud computing outsourcing has attracted considerable research efforts both from theoretical cryptographers and security engineers. The theoretical cryptography commu-

TABLE 3. Simulation experiment results.

Dimension	$t_{DFT}(msec)$	$t_{radix-2\ FFT}(msec)$	$t_{outsourcing}(msec)$	$\frac{t_{DFT}}{t_{outsourcing}}(msec)$	$\frac{t_{radix-2\ FFT}}{t_{outsourcing}}(msec)$
256	2.6978	0.0466	0.0532	50.7105	0.8759
512	10.2014	0.0963	0.0997	102.3210	0.9659
1024	43.6563	0.2079	0.1804	241.9972	1.1524
2048	160.6058	0.4689	0.3720	431.7360	1.2605
5096	680.7852	1.3077	0.9483	717.9007	1.3790
10192	2608.5201	2.9276	1.9475	1339.4198	1.5033

nity usually focus on designing generic protocols covering all problems, mainly based on Yao's garbled circuits [16], [17] and Gentry's fully homomorphic encryption (FHE) scheme [18]. Their basic idea of encryption is to model any polynomial time computation using a circuit, for example, [19], [20]. Due to the extremely high complexity of the FHE operation and the pessimistic circuit sizes, the generic protocols are quite complicated and inefficient that seems to be far from practical. In contrast, the security engineering community typically focuses on designing different outsourcing protocols for specific problems. Early outsourcing protocols usually allowed privacy leaks and the verification of returned results were also ignored, for example, [21]–[23]. Later, the malicious cloud model was accepted by most outsourcing protocols and proposed a series of secure and efficient outsourcing algorithms. Among them, there are the outsourcing of matrix inversion [24], the outsourcing of linear programming [25], and the outsourcing of support vector machine (SVM) [26]. Recently, efforts have been made to apply cloud computing to specific real-world scenarios, and the corresponding research results appeared in [27]–[29].

The fast algorithm for DFT is an important research topic in the field of signal processing. Since Cooley *et al.* proposed the radix-2 FFT algorithm [13], there are many fast algorithms that have been proposed, such as split-radix FFT [30], discrete Hartley transform (DHF) [31], radix-4 FFT [32], radix-r FFT [33], and mixed-radix FFT [34]. Among them, split-radix FFT algorithm has the highest efficiency, and its complex multiplication times are close to the theoretical minimum value of FFT. Moreover, new improved algorithms are still being proposed. In the encrypted domain, Bianchi *et al.* investigated the implementation of the DFT on a vector of encrypted samples [35], [36] and the implementation of the FFT on a vector of encrypted samples [37], both of which rely on the homomorphic properties of the underlying cryptosystem. However, due to the huge computational complexity of homomorphic encryption scheme, they are not suitable for practical applications. Instead, from the perspective of security engineering community, the proposed outsourcing protocol for DFT, IDFT, and circular convolution pay more attention to the efficiency and therefore are closer to practical applications.

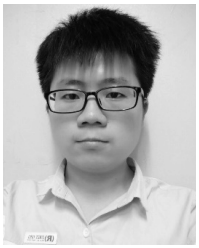
VII. CONCLUSIONS

In this paper, we have presented a secure and efficient cloud computing outsourcing protocol for DFT, IDFT, and circular convolution. The proposed protocol successfully implements the design goals under the malicious cloud model. Through theoretical analysis and simulation experiments, it is found that as long as N is not too small, cloud computing has more obvious efficiency advantages than FFT algorithm. Moreover, unlike FFT, cloud computing does not impose any restrictions on the value of N. In addition, the trade-offs between security, efficiency, and accuracy can be flexibly adjusted depending on usage. Therefore, in an application scenario with cloud computing conditions, using computing outsourcing to speed up the operation of local signal processing equipment is also an alternative method.

REFERENCES

[1] M. Frerking, *Digital Signal Processing in Communications Systems*. Medford, MA, USA: Springer, 2013.
[2] R. Martin and C. Breithaupt, "Speech enhan cement in the DFT domain using Laplacian speech priors," in *Proc. Int. Workshop Acoustic Echo Noise Control (IWAENC)*, vol. 3, 2003, pp. 87–90.
[3] S. Erturk and T. J. Dennis, "Image sequence stabilisation based on DFT filtering," *IEE Proc.-Vis., Image Signal Process.*, vol. 147, no. 2, pp. 95–102, Apr. 2000.
[4] Z. Pan, H. Qin, X. Yi, Y. Zheng, and A. Khan, "Low complexity versatile video coding for traffic surveillance system," *Int. J. Sensor Netw.*, vol. 30, no. 2, pp. 116–125, 2019.
[5] J. A. Berger, S. K. Mitra, and J. Astola, "Power spectrum analysis for DNA sequences," in *Proc. 7th Int. Symp. Signal Appl. (ISSPA)*, vol. 2, Jul. 2003, pp. 29–32.
[6] Y. H. Kim, I. Song, H. G. Kim, T. Chang, and H. M. Kim, "Performance analysis of a coded OFDM system in time-varying multipath Rayleigh fading channels," *IEEE Trans. Veh. Technol.*, vol. 48, no. 5, pp. 1610–1615, Sep. 1999.
[7] B.-C. Wang, *Digital Signal Processing Techniques and Applications in Radar Image Processing*, vol. 91. Hoboken, NJ, USA: Wiley, 2008.
[8] K. R. Gledhill, "An earthquake detector employing frequency domain techniques," *Bull. Seisoml. Soc. Amer.*, vol. 75, no. 6, pp. 1827–1835, 1985.
[9] D. Wubben *et al.*, "Benefits and impact of cloud computing on 5G signal processing: Flexible centralization through cloud-RAN," *IEEE Signal Process. Mag.*, vol. 31, no. 6, pp. 35–44, Nov. 2014.
[10] C. Zhu, L. Shu, T. Hara, L. Wang, S. Nishio, and L. T. Yang, "A survey on communication and data management issues in mobile sensor networks," *Wireless Commun. Mobile Comput.*, vol. 14, no. 1, pp. 19–36, Jan. 2014.
[11] X. Zhai, X. Guan, C. Zhu, L. Shu, and J. Yuan, "Optimization algorithms for multiaccess green communications in Internet of Things," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1739–1748, Jun. 2018.
[12] Z. Sheng, C. Mahapatra, C. Zhu, and V. C. M. Leung, "Recent advances in industrial wireless sensor networks toward efficient management in IoT," *IEEE Access*, vol. 3, pp. 622–637, 2015.
[13] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Comput.*, vol. 19, no. 90, pp. 297–301, 1965.
[14] J. G. Proakis and D. K. Manolakis, *Digital Signal Processing*, 4th ed. 2006.
[15] Y. Lindell, "Secure multiparty computation for privacy preserving data mining," in *Encyclopedia Data Warehousing Mining*. New York, NY, USA: IGI Global, 2005, pp. 1005–1009.

[16] A. C.-C. Yao, "Protocols for secure computations," in *Proc. 23rd Annu. Symp. Found. Comput. Sci.*, vol. 82, Nov. 1982, pp. 160–164.
[17] A. C.-C. Yao, "How to generate and exchange secrets," in *Proc. IEEE Symp. Found. Comput. Sci. (FOCS)*, Oct. 2008, pp. 162–167.
[18] C. Gentry and D. Boneh, *A Fully Homomorphic Encryption Scheme*, vol. 20, no. 9. Stanford, CA, USA: Stanford Univ., 2009.
[19] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive verifiable computing: Outsourcing computation to untrusted workers," in *Proc. 30th Ann. Conf. Adv. Cryptol. (CRYPTO)*. Berlin, Germany: Springer, 2010, pp. 465–482.
[20] K.-M. Chung, Y. Kalai, and S. Vadhan, "Improved delegation of computation using fully homomorphic encryption," in *Proc. 30th Ann. Conf. Adv. Cryptol. (CRYPTO)*. Berlin, Germany: Springer, 2010, pp. 483–501.
[21] M. J. Atallah and K. B. Frikken, "Securely outsourcing linear algebra computations," in *Proc. 5th ACM Symp. Inf. Comput. Commun. Secur.*, 2010, pp. 48–59.
[22] D. Benjamin and M. J. Atallah, "Private and cheating-free outsourcing of algebraic computations," in *Proc. IEEE 6th Annu. Conf. Privacy, Secur. Trust*, Oct. 2008, pp. 240–245.
[23] M. J. Atallah, K. N. Pantazopoulos, J. R. Rice, and E. E. Spafford, "Secure outsourcing of scientific computations," *Adv. Comput.*, vol. 54, pp. 215–272, 1998.
[24] X. Lei, X. Liao, T. Huang, H. Li, and C. Hu, "Outsourcing large matrix inversion computation to a public cloud," *IEEE Trans. Cloud Comput.*, vol. 1, no. 1, p. 1, Jan./Jun. 2013.
[25] C. Wang, K. Ren, and J. Wang, "Secure optimization computation outsourcing in cloud computing: A case study of linear programming," *IEEE Trans. Comput.*, vol. 65, no. 1, pp. 216–229, Jan. 2016.
[26] X. Li, Y. Zhu, J. Wang, Z. Liu, Y. Liu, and M. Zhang, "On the soundness and security of privacy-preserving SVM for outsourcing data classification," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 5, pp. 906–912, Sep./Oct. 2018.
[27] M. R. Sarker, J. Wang, Z. Li, and K. Ren, "Security and cloud outsourcing framework for economic dispatch," *IEEE Trans. Smart Grid*, vol. 9, no. 6, pp. 5810–5819, Nov. 2017.
[28] Y. Zhang *et al.*, "Computation outsourcing meets lossy channel: Secure sparse robustness decoding service in multi-clouds," *IEEE Trans. Big Data*, to be published.
[29] Y. Zhang, J. Jiang, Y. Xiang, Y. Zhu, L. Wan, and X. Xie, "Cloud-assisted privacy-conscious large-scale Markowitz portfolio," *Inf. Sci.*, to be published. doi: 10.1016/j.ins.2018.12.055.
[30] P. Duhamel and H. Hollmann, "'Split radix' FFT algorithm," *Electron. Lett.*, vol. 20, no. 1, pp. 14–16, 1984.
[31] R. N. Bracewell, "Discrete Hartley transform," *J. Opt. Soc. Amer.*, vol. 73, no. 12, pp. 1832–1835, 1983.
[32] G. H. Allen, "Programming an efficient radix-four FFT algorithm," *Signal Process.*, vol. 6, no. 4, pp. 325–329, 1984.
[33] M. A. Jaber and D. Massicotte, "The radix-r one stage FFT kernel computation," in *Proc. IEEE Int. Conf. Acoust. (ICA)*, Mar./Apr. 2008, pp. 3585–3588.
[34] B. M. Xiang, X. W. Jin, and L. Y. Hai, "A variable-length FFT processor base on mixed-radix algorithm for PAPR reduction in OFDM systems," *Adv. Mater. Res.*, vols. 588–589, pp. 826–829, Nov. 2012.
[35] T. Bianchi, A. Piva, and M. Barni, "Implementing the discrete Fourier transform in the encrypted domain," in *Proc. IEEE Int. Conf. Acoust. (ICA)*, Mar./Apr. 2008, pp. 1757–1760.
[36] T. Bianchi, A. Piva, and M. Barni, "On the implementation of the discrete Fourier transform in the encrypted domain," *IEEE Trans. Inf. Forensics Security*, vol. 4, no. 1, pp. 86–97, Mar. 2009.
[37] T. Bianchi, A. Piva, and M. Barni, "Comparison of different FFT implementations in the encrypted domain," in *Proc. Eur. Signal Conf. (EUSIPCO)*, 2008.



XIANGLI XIAO is currently pursuing the degree with the School of Electronics and Information Engineering, Southwest University, Chongqing, China. His current research interests include multimedia security, cloud computing security, big data security, and the Internet of Things security.

JUNJIAN HUANG received the B.S. degree from the Chongqing Communication Institute, Chongqing, China, in 2002, and the Ph.D. degree in computer science and technology from Chongqing University, Chongqing, in 2014. He has been a Professor with the Chongqing University of Education, Chongqing, since 2014. His current research interests include neural networks, memristive systems, intermittent control and synchronization, and information security.

YUSHU ZHANG (M'17) received the Ph.D. degree from the College of Computer Science, Chongqing University, Chongqing, China, in 2014. He held various research positions with Southwest University, the City University of Hong Kong, the University of Macau, and Deakin University. He is currently a Professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China. His research interests include multimedia security, compressive sensing security, cloud computing, and big data security. He has published over 80 refereed journal articles and conference papers in these areas.

XING HE received the B.S. degree in mathematics and applied mathematics from the Department of Mathematics, Guizhou University, Guiyang, China, in 2009, and the Ph.D. degree in computer science and technology from Chongqing University, Chongqing, China, in 2013. From 2012 to 2013, he was a Research Assistant with Texas A&M University at Qatar, Doha, Qatar. He is currently a Professor with the School of Electronics and Information Engineering, Southwest University, Chongqing. His research interests include neural networks, bifurcation theory, optimization method, smart grid, and nonlinear dynamical systems.