Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 8158465, 8 pages
https://doi.org/10.1155/2017/8158465

Research Article

Semitensor Product Compressive Sensing for Big Data Transmission in Wireless Sensor Networks

Haipeng Peng,[1,2] Ye Tian,[1,2] and Jürgen Kurths[3]

[1]*Information Security Center, State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*National Engineering Laboratory for Disaster Backup and Recovery, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[3]*Potsdam Institute for Climate Impact Research, Potsdam 14473, Germany*

Correspondence should be addressed to Haipeng Peng; penghaipeng@bupt.edu.cn

Received 16 January 2017; Accepted 9 March 2017; Published 22 March 2017

Academic Editor: Liu Yuhong

Big data transmission in wireless sensor network (WSN) consumes energy while the node in WSN is energy-limited, and the data transmitted needs to be encrypted resulting from the ease of being eavesdropped in WSN links. Compressive sensing (CS) can encrypt data and reduce the data volume to solve these two problems. However, the nodes in WSNs are not only energy-limited, but also storage and calculation resource-constrained. The traditional CS uses the measurement matrix as the secret key, which consumes a huge storage space. Moreover, the calculation cost of the traditional CS is large. In this paper, semitensor product compressive sensing (STP-CS) is proposed, which reduces the size of the secret key to save the storage space by breaking through the dimension match restriction of the matrix multiplication and decreases the calculation amount to save the calculation resource. Simulation results show that STP-CS encryption can achieve better performances of saving storage and calculation resources compared with the traditional CS encryption.

1. Introduction

For its ease of deployment and cost effectiveness, wireless sensor network (WSN) is widely used in environment monitoring, disaster relief, military, and so on [1–3]. For example, with WSN for forest fire monitoring, numerous sensors are deployed in the monitor area, and big data should be gathered and transmitted in real-time. Big data transmission consumes vast energy, but the node in WSNs is energy-limited. If the battery of the node is drained, the node is useless. Moreover, since the nodes in WSNs use wireless communication technologies, the link is easy to be eavesdropped, and the data transmission needs to be encrypted. So big data transmission in WSNs should solve the energy-efficiency and encryption problems. CS can decrease the volume of the data transmitted, which can save the energy to prolong the life of the node [4, 5]. CS is also a kind of encryption method resulting from the randomness of the measurement matrix [6]. So CS can be used to encrypt data and save energy simultaneously.

However, the CS encryption uses the measurement matrix as the secret key, while the measurement matrix needs a huge storage space which is not suitable for WSNs. In WSNs, nodes are not only energy-limited, but also storage and calculation resources-constrained. Many optimization methods for the measurement matrix have been proposed [7, 8]. However, most of these existing methods focused on how to improve the recovery accuracy, decrease the iteration number, and accelerate the calculation. For reducing the size of the measurement matrix, most works focused on reducing the row number of the measurement matrix [9, 10], because the column number must be equal to the signal length according to the rule of matrix multiplication. Another kind of method to reduce the matrix size is dividing the signal into blocks, but this needs extra data processing overhead.

Another kind of CS encryption can save storage space by storing matrix generation parameters as the secret key rather than the whole matrix [11, 12]. This kind of CS encryption generates matrices by deterministic methods such

as algebraic curves [13], coding (LDPC, BCH) [14], and chaotic systems (Chebyshev, Logistic, and Tent) [15, 16]; it can save huge storage space compared with keeping the whole matrix. However, using this method, users have to generate the matrix before encryption for each transmission. Although the deterministic method can decrease the key storage space by storing parameters, it needs to calculate the measurement matrix in real-time, which is at the expense of the calculation resource.

In this paper, semitensor product compressive sensing (STP-CS) is proposed to solve the problems above. STP-CS can save the storage space by introducing the semitensor product [17–19] into compressive sensing, which can break through the dimension restriction of matrix multiplication and reduce the row and column numbers of the measurement matrix simultaneously. Compared with deterministic methods, the calculation resource of STP-CS is saved, because STP-CS does not need to generate the matrix in real-time before data encryption. An algorithm for STP-CS is also proposed, which saves the calculation resource compared with the traditional CS in theory and under simulation. Contributions of this paper are as follows:

(i) STP-CS reduces the row and column numbers of the measurement matrix simultaneously to save the storage space.

(ii) An algorithm of STP-CS is proposed to save the computing resources.

(iii) The recovery performance of STP-CS is similar to those of the traditional CS and CCS, and the compression ratio performance of STP-CS is not affected.

The rest of this paper is organized as follows. Section 2 introduces the details of STP-CS encryption. The storage and calculation resources of STP-CS are analyzed in Section 3. Simulation results are discussed in Section 4. The last section concludes this paper.

Notation. The following notation is used throughout the paper. WSN denotes the wireless sensor network. CS denotes compressive sensing. CCS denotes the chaotic compressive sensing. STP-CS denotes the semitensor product compressive sensing. x, y denote the plain message and cipher message, respectively. P, K are the length and sparsity of x, respectively. Φ denotes the measurement matrix, and M, N are the row and column number of the measurement matrix, respectively.

2. Related Works

In this section, some works about how to decrease the storage space of the CS secret key are introduced. There are two kinds of methods to decrease the storage space; one kind is reducing the size of the measurement matrix. Another is using the deterministic measurement matrix, and with this method, the matrix generation parameters are saved rather than the whole matrix.

A method for designing the measurement matrix is proposed in [10]. This method can reduce the row number of the measurement matrix, but the side information is needed for the design of the measurement matrix. Model based compressive sensing is proposed in [9]. Using this model based CS, the signal can be recovered by less number of measurements by leveraging more realistic signal models; less number of measurements means the row number of measurement matrix is reduced. But the recovery algorithm has to be improved; the traditional recovery algorithm cannot be used. Compared with these methods, STP-CS can reduce not only the row number of the measurement matrix, but also the column number by breaking through the restriction of matrix multiplication.

There are many kinds of deterministic measurement matrices. The chaotic sequence has the property of pseudo-random, so it can be used for constructing the measurement matrix [15]. The possibility of constructing measurement matrix with different kinds of chaotic systems is investigated, including Chen system, Chua system, and Lorenz system [16]. Algebraic curves like elliptic curves can also be used to construct the deterministic measurement matrices [13]. LDPC code is another kind of method for constructing the deterministic measurement matrices [14]. All these methods only need to store some parameters rather than the whole matrix, but the measurement matrix has to be generated in real-time. Compared with these methods, STP-CS can save huge computing resource.

In addition, there are many other matrices which can be used as deterministic measurement matrix, such as cyclic matrix [20], Toeplitz matrix [21], chirp matrix [22], and polynomial matrix [23]. However, these matrices have other restrictions. Cyclic matrix and Toeplitz matrix still need to store lots of test data, and the construction of polynomial matrix is limited by the signal length [20].

3. STP-CS Data Communication

In this section, the details of our proposed STP-CS encryption are introduced. Before this, CS encryption is introduced.

3.1. CS Encryption. Based on CS theory [24, 25], suppose $x \in R^N$ is a plain message; project x to $y \in R^M$ using the matrix $\Phi \in R^{M\times N}$, $y = \Phi x$, where Φ is called the measurement matrix and $M < N$. Because y is very different from x, y is regarded as the cipher message, and Φ is the secret key. At the receiver, x can be recovered with y and Φ by utilizing some algorithms such as BP, OMP, and ROMP [24, 26, 27]. For the recovery, x should be sparse or sparse on some orthogonal basis $\Psi \in R^{N\times N}$; that is, $x = \Psi s$. The sparsity here means K values of s are nonzero, while the other $N - K$ values are zero, where $K \ll N$. Though $M < N$, for the accuracy recovery, M cannot be arbitrarily small; it has to be satisfied with

$$M \geq cK \log_2\left(\frac{N}{K}\right), \tag{1}$$

where c is a small constant [24]. Resulting from the dimension restriction of the matrix multiplication, the column number N of the measurement matrix has to be equal to the dimension of the signal x. For storing a CS secret key, MN elements

need to be stored. So, to decrease the size of the measurement matrix, this restriction has to be broken through.

3.2. Semitensor Product. The semitensor product (STP) was proposed by Cheng and Zhang in [17]. STP is the generalization of the conventional matrix multiplication, and it can break through the dimension match restriction of the conventional matrix multiplication.

Suppose u is a row vector of dimension np; v is a column vector of dimension p; dividing u to p equal parts, that is, $u^1, \ldots, u^p$, each part u^i is a row vector of dimension n. The definition of STP, denoted by $\ltimes$, is

$$u \ltimes v = \sum_{i=1}^{p} u^i v_i \in R^{1\times n}. \tag{2}$$

Similarly, $v^T \ltimes u^T = \sum_{i=1}^{p} v_i (u^i)^T \in R^{n\times 1}$. Generalized to a matrix, suppose $A \in R^{m\times n}$, $B \in R^{p\times q}$; if n is the factor of p or p is the factor of n, the definition of the semitensor product of A and B is as follows:

$$A \ltimes B = \begin{bmatrix} A_1 \ltimes B^1 & \cdots & A_1 \ltimes B^q \\ \vdots & \ddots & \vdots \\ A_m \ltimes B^1 & \cdots & A_m \ltimes B^q \end{bmatrix}, \tag{3}$$

where A_i denotes the ith row of A and B^j denotes the jth column of B.

3.3. STP-CS Encryption. Now, introduce STP into CS encryption [28]. The definition of STP-CS is as follows:

$$y = A \ltimes x, \tag{4}$$

where $A \in R^{M\times N}$, $M < N$, $x \in R^P$. To decrease the size of the measurement matrix A, N should be as small as possible. For meeting the requirement of STP, we choose N with the condition $N \mid P$. According to [17],

$$y = A \ltimes x = \left(A \otimes I_{P/N}\right) x, \tag{5}$$

where $y \in R^{MP/N}$ and $\otimes$ denotes the Kronecker product [17]. When $N = P$, (5) translates to $y = Ax$, which is the traditional CS. From (5), STP-CS with the measurement matrix A is equivalent to the traditional CS with the measurement matrix $(A \otimes I_{P/N})$. The RIP, spark, and coherence of the measurement matrix are introduced in [28], A needs to meet these conditions. Based on the definition of STP-CS, for a signal $x \in R^P$, the column number of the measurement matrix A only needs to be satisfied with the condition $N \mid P$, while the traditional CS should meet the dimension match, and the column number must be equal to P. So compared with the traditional CS, the column number of the measurement matrix can be decreased. As for the row number of the measurement matrix in STP-CS, it can be also decreased which will be introduced in next section, while the row number of the measurement matrix in the traditional CS cannot break through the restriction in (1). So, although the STP-CS encryption also keeps the measurement matrix as the secret key, it can save a huge storage space by decreasing the size of the measurement matrix.

Compared with deterministic methods [15], like chaotic compressive sensing (CCS), the storage space of the measurement matrix in STP-CS is not decreased, because CCS stores matrix generation parameters such as chaotic parameter or chaos sequence initial value. But calculation resource of STP-CS is saved. The measurement matrix in CCS has to be generated in real-time, which will need much calculation resource. So STP-CS can save storage space compared with the traditional CS and save calculation resource compared with CCS. In fact, STP-CS can also save calculation resource compared with the traditional CS, which will be introduced in the next section. So STP-CS can be widely used in resource-limited scenarios like WSNs. STP-CS encryption can not only solve the security and energy-efficiency problems but also save storage and calculation resources.

3.4. An Algorithm for STP-CS. In this part, an algorithm for STP-CS is proposed, which can implement STP-CS using less calculation resource than the traditional CS.

From (3), computing $A \ltimes x$ needs to compute $A_i \ltimes x$, $i = 1, 2, \ldots, M$. To compute $A_i \ltimes x$, split x to P/N, and use each element of A_i to multiply the corresponding block of x, which means that every element of A needs to be multiplied by several numbers. As for matrix multiplication, suppose $\widetilde{C} = \widetilde{A}\widetilde{B}$; an arbitrary element $\widetilde{a}_{ij}$ of $\widetilde{A}$ needs to be multiplied by every element of the jth row of $\widetilde{B}$. Based on the above analysis, an algorithm for STP-CS using matrix multiplication is proposed.

(1) Project x to an $N \times (P/N)$ matrix as follows:

$$x_{\text{matrix}} = \begin{bmatrix} x_1 & x_2 & \cdots & x_{P/N} \\ x_{1+P/N} & x_{2+P/N} & \cdots & x_{2P/N} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1+P(N-1)/N} & x_{2+P(N-1)/N} & \cdots & x_{N\times P/N} \end{bmatrix}. \tag{6}$$

(2) Left multiply the above matrix x_{matrix} using the STP-CS measurement matrix A:

$$y_{\text{matrix}} = A x_{\text{matrix}}. \tag{7}$$

(3) Transform each row of y_{matrix} into a column vector, and construct a new column vector using these vectors. This new column vector is equal to y.

Next is the brief proof for step (3). Based on the second step of the algorithm, we have

$$y_{\text{matrix}}(ij) = \sum_{l=1}^{N} a_{il} x_{j+(l-1)P/N}. \tag{8}$$

And y can be split into M blocks with P/N elements; the jth element of the ith block of y is

$$y_i^j = \sum_{l=1}^{N} a_{il} x_{j+(l-1)P/N}. \tag{9}$$

So $y_i^j = y_{\text{matrix}}(ij)$, and y_{matrix} can be transformed to y.

The diagram of the STP-CS algorithm above is shown in Figure 1. x is the plain message and y is the cipher message. A is the secret key of STP-CS encryption.

4. Performance Analysis

In this section, the performance of STP-CS is analyzed, including storage resource, calculation resource, and compression ratio.

Based on (5), STP-CS with the measurement matrix $A \in R^{M\times N}$ is equivalent to the traditional CS with the measurement matrix $(A \otimes I_{P/N}) \in R^{(MP/N)\times P}$. According to (1), we have

$$\frac{MP}{N} \geqslant cK\log_2\left(\frac{P}{K}\right). \tag{10}$$

And then

$$M \geqslant \frac{cNK\log_2(P/K)}{P}, \tag{11}$$

where c is a small constant. In order to compress the signal, the dimension of y should be satisfied with $MP/N < P$; that is, $M < N$, so the range of the row number of A is

$$\frac{cNK\log_2(P/K)}{P} \leqslant M < N. \tag{12}$$

Because storing a measurement matrix needs to keep MN elements, the range of the storage space for one STP-CS key is

$$\frac{cN^2K\log_2(P/K)}{P} \leqslant MN < N^2. \tag{13}$$

Based on $N \mid P$, set $P = kN$, $k \in Z^+$, and k is the factor of P; (13) can be transformed to

$$\frac{cPK\log_2(P/K)}{k^2} \leqslant MN < \frac{P^2}{k^2}. \tag{14}$$

Equation (14) is the relationship between the key storage space and the dimension and sparsity of the signal x. For comparison, suppose $A' \in R^{M'\times N'}$ is the measurement matrix for the traditional CS. To encrypt the same signal, the condition $N' = P$, $M' \geq cK\log_2(P/K)$ should be satisfied, so the storage space for one traditional CS key is

$$cPK\log_2\left(\frac{P}{K}\right) \leqslant M'N' < P^2. \tag{15}$$

From (14) and (15), the low bound of one STP-CS key storage space is smaller than that of one traditional CS key storage space, when $k \neq 1$.

According to (4) and (5), y is a column vector of dimension MP/N. According to (3), y includes M column vectors with dimension P/N; that is,

$$y = \begin{bmatrix} y_1 & y_2 & \cdots & y_M \end{bmatrix}^T, \tag{16}$$

where $y_i = a_i \ltimes x = \sum_{j=1}^{N} a_{ij}x^j$, in which a_i is the ith row of A, and x^j is jth block of x. The dimension of x^j is P/N. From (2), computing each y_i needs $(P/N) \times N$ multiplications, that is, P multiplications, and $(N-1)\times(P/N)$ additions. y includes M y_i. So computing the whole y needs MP multiplications and $(N-1)MP/N$ additions. However, using the traditional CS needs the measurement matrix $A' \in R^{(MP/N)\times P}$ in order to get the same data volume of STP-CS. Computing each measurement needs P multiplications and $P-1$ additions. Computing the whole y needs MP^2/N multiplications and $MP(P-1)/N$ additions. To get the same number of measurements, the multiplication resources of the traditional CS are P/N times that of STP-CS; the addition resources of traditional CS are $(P-1)/(N-1)$ times that of STP-CS. Resulting from $N \mid P$, $P/N \geqslant 1$, the traditional CS needs more resources than STP-CS. When $P = N$, the resources needed are the same for both methods. In fact, if $P = N$, STP-CS degenerates to the traditional CS.

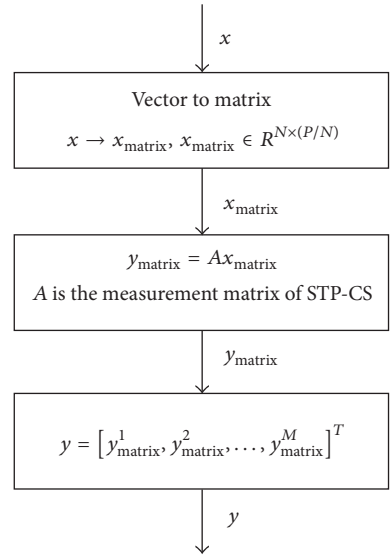

Figure 1: STP-CS algorithm diagram.

Now analyze the computing resource of the algorithm proposed in Section 2. Computing each element of y_{matrix} needs N multiplications and $N-1$ additions, and y_{matrix} has MP/N elements, so the whole resources needed for computing y_{matrix} are MP multiplications and $MP((N-1)/N)$ additions. Compared with the definition of STP-CS, the calculation quantity is the same.

Next, we analyze the compression ratio of STP-CS. From (5), the dimension of y is MP/N, so the compression ratio is $R = (MP/N)/P = M/N$, where M and N are the row number and column number of the STP-CS measurement matrix, respectively. From (12), the range of compression ratio R of STP-CS is $cK\log(P/K)/P \leqslant M/N < 1$. And the compression ratio of the traditional CS is $R' = M'/P$, where M' is the row number of the traditional CS measurement matrix, and P is the dimension of the signal. From the row number restriction, the range of R' of traditional CS is $(cK\log(P/K))/P \leqslant M'/P < 1$. So the range of STP-CS is the same as that of the traditional CS. STP-CS can obtain the same compression ratio as the traditional CS.

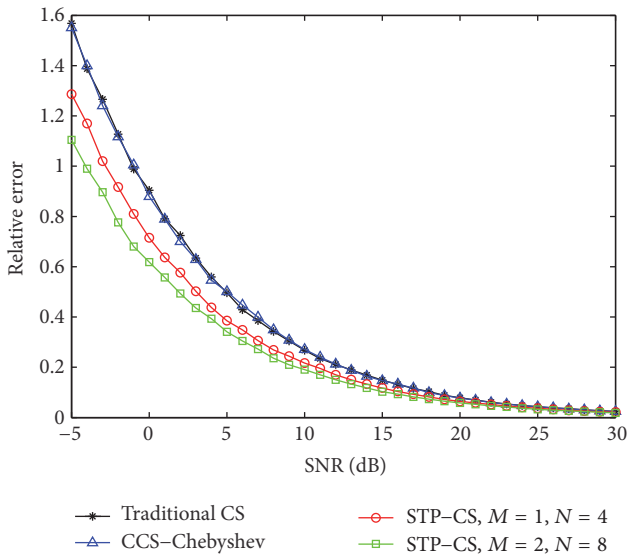

Figure 2: Recovery results. The range of SNR is from −5 dB to 30 dB. The traditional CS uses a Gaussian matrix, CCS uses a Chebyshev matrix, and STP-CS uses a Gaussian matrix. The simulation time is 200; the relative errors are the average values of the 200 simulation results.

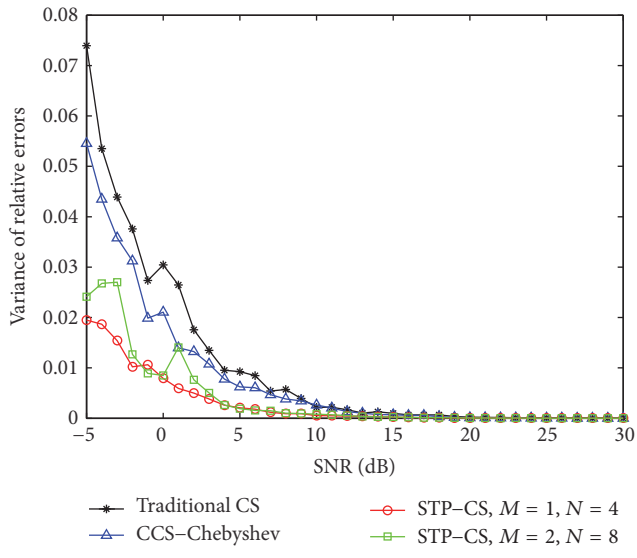

Figure 3: Variance of the relative errors of 200 simulation results. The range of SNR is from −5 dB to 30 dB. The traditional CS uses a Gaussian matrix, CCS uses a Chebyshev matrix, and STP-CS uses a Gaussian matrix.

5. Simulation Results

In this section, simulations of STP-CS encryption and decryption are discussed. In the experiment, the length of the original signal x is 256, and the sparsity K is 7. The signal is a frequency domain sparse signal, which is combined by some discrete sine signals. The recovery algorithm is OMP, and the recovery performance is measured by the relative error,

$$\delta = \frac{\|\widetilde{x} - x\|_2}{\|x\|_2}, \tag{17}$$

which is the 2-norm of the recovery error $\widetilde{x} - x$ relative to the 2-norm of the original signal x, and $\widetilde{x}$ is the recovered signal. The simulation time is 200, and the relative errors in Figures 2, 5, and 6 are the average values of the 200 simulation results.

Figure 2 shows the recovery performance of STP-CS compared with the traditional CS and CCS [16]. The compression ratio is 0.25, and two groups of the STP-CS matrix M, N are processed. The sizes of two STP-CS matrices are $M = 1$, $N = 4$ and $M = 2$, $N = 8$, respectively. The sizes of the traditional CS and CCS are both 64×256. From Figure 2, four curves coincide with each other after 20 dB. For example, At 20 dB, the relative error of the traditional CS is 0.0788, the relative error of CCS is 0.0769, the relative error of STP-CS with 1×4 matrix is 0.0644, the relative error of STP-CS with 2×8 matrix is 0.0579, and the relative errors of three kinds of matrices tend to be zero at 30 dB. Figure 3 shows the variance of the relative errors of the 200 simulation results. From Figure 3, the variance is small, after 10 dB, all variances are less than 0.01. The variance of STP-CS is smaller than the traditional CS and CCS. So the relative error of STP-CS is stable. This implies that the recovery performance of STP-CS is similar to those of the traditional CS and CCS. The size of the STP-CS matrix is extremely small, which implies that the small-size measurement matrix of STP-CS achieves a similar recovery performance as that of the traditional CS, so STP-CS can save storage space.

Not only the above signal but also STP-CS can be used for other kinds of signals. Table 1 shows the recovery results of four kinds of signals. These signals are generated by MATLAB, including Bernoulli, Gaussian, Uniform, and Power distributions. The recovery algorithm is OMP, and the measurement matrix is a 16×32 Gaussian matrix for three kinds of length signals. From Table 1, the relative errors are small for these four kinds of signals. STP-CS can encrypt the signals with different length, but the dimension of the measurement matrix of the traditional CS should be adjusted to match the signal length.

STP-CS can be also used for the image. Figure 4 shows the recovery results for a Lena image. The image size is 512×512, the size of the measurement matrix is 64×128, and the PSNR (Peak Signal to Noise Ratio) is 33.64 dB. The compression ratio of STP-CS is 0.5. For the traditional CS, the size of the measurement matrix should be 256×512 for the compression ratio 0.5, so STP-CS can save huge storage space for the measurement matrix.

The computing resources are measured by computing time. In this part, the encryption time is recorded by MATLAB system time, and the unit is millisecond. The compression ratio is also 0.25. The size of the STP-CS matrix is 1×4, the size of the traditional CS matrix is 64×256, and the size of the chaotic matrix is 64×256. The encryption time of the above three methods is 0.161 ms, 0.254 ms, and 1.953 s, respectively. Because the CCS needs to generate the matrix, the time of CCS is very long. Table 1 shows the computing time for different groups of M, N for the STP-CS matrix. From Table 2, the computing time increases with the

Table 1: Relative error of the recovery results for different signals. The encryption method is STP-CS.

Length	Signal			
	Bernoulli	Gaussian	Power	Uniform
256	2.33×10^{-16}	1.06×10^{-16}	3.10×10^{-16}	9.77×10^{-16}
512	2.42×10^{-16}	1.01×10^{-16}	3.20×10^{-16}	1.53×10^{-16}
1024	1.64×10^{-16}	2.49×10^{-16}	1.56×10^{-16}	6.31×10^{-17}

(a) (b)

Figure 4: STP-CS for image. (a) is the original image. (b) is the recovery image. The measurement matrix is a Gaussian matrix, and the recovery algorithm is OMP.

Table 2: Encryption times for different groups of M, N. The encryption method is STP-CS, unit ms.

Signal length	Matrix			
	1×4	1×8	4×16	4×32
256	0.161	0.163	0.172	0.175
512	0.163	0.163	0.171	0.172
1024	0.165	0.168	0.174	0.177
2048	0.166	0.171	0.187	0.195

increments of M, N, and P. So, to reduce the computing time, M and N should be small. Along with the increment of the signal length, the traditional CS should increase the row of the measurement matrix which will increase the calculation quantity, while the STP-CS does not need to increase the row number. So the STP-CS also has the advantage on the decrement of computing resources.

In Figure 5, the compression ratio performance of STP-CS is shown. The compression ratio of STP-CS is M/N; to get the small-size matrix, we choose M, N as small as possible. At 20 dB, the relative error of the 1×8 matrix is 0.0840, the relative error of the 1×4 matrix is 0.0655, the relative error of the 2×4 matrix is 0.0461, and the relative error of the 3×4 matrix is 0.0374. The recovery errors of these four matrices tend to be zero after 20 dB. This implies that the STP-CS can also achieve low compression ratio without affecting the recovery accuracy. Even the compression ratio is 0.125; at 30 dB, the relative error is 0.0261, similar to that of the ratio of 0.25, 0.0207. So the relative error can also tend to be zero at high SNR.

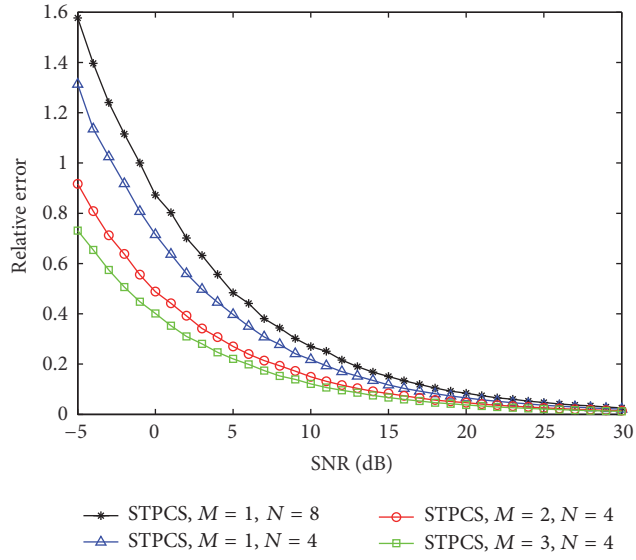

Figure 5: Performance of compression ratio. Four kinds of ratios, that is, 0.125, 0.25, 0.5, and 0.75, are tested. The simulation time is 200; the relative errors are the average values of the 200 simulation results.

From Figure 6, only the original matrix can decrypt the data correctly, and the recovery errors of the other three

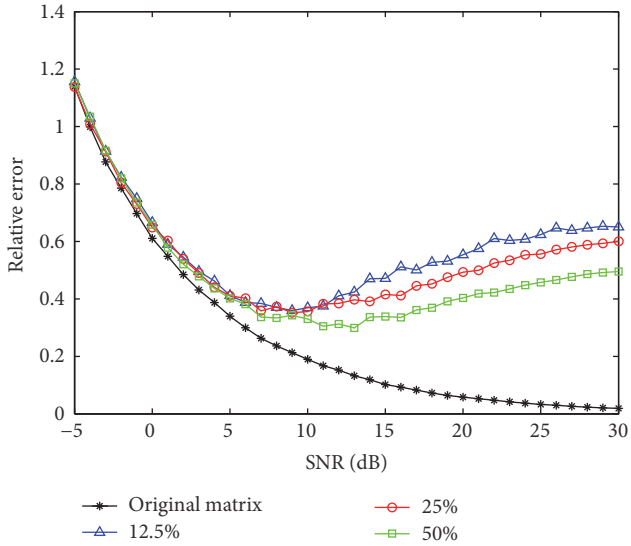

Figure 6: Security of STP-CS encryption. The data are encrypted by a 2×8 matrix. Four matrices are used to decrypt the encryption data, including the original matrix, 12.5% of the elements the same as the original matrix, 25% of the elements the same as the original matrix, and 50% of the elements the same as the original matrix. The left unknown elements are generated randomly.

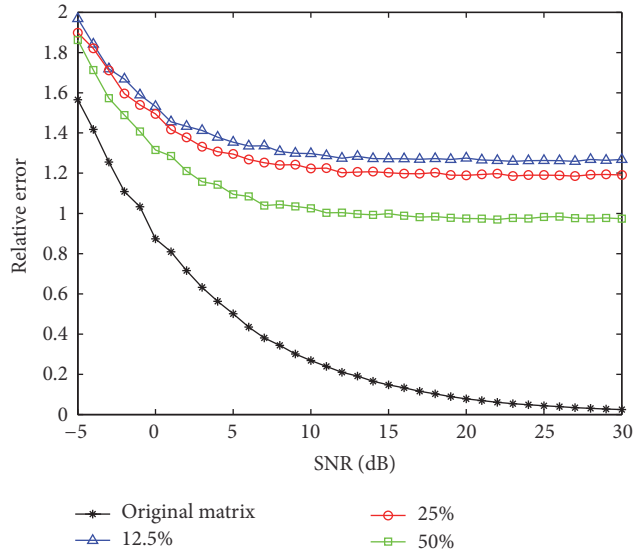

Figure 7: Security of the traditional CS encryption. The data are encrypted by a 64×256 matrix. Four matrices are used to decrypt the encryption data, including the original matrix, 12.5% of the elements the same as the original matrix, 25% of the elements the same as the original matrix, and 50% of the elements the same as the original matrix. The left unknown elements are generated randomly.

matrices are larger than 20% from −5 dB to 30 dB. The elements of other three matrices are only partly the same as those of the original matrix, and the encrypted data cannot be decrypted by a different key. Even if there is an eavesdropper who has 50% of the elements of the key, the encrypted data still cannot be decrypted. At 30 dB, the relative errors of these three matrices are larger than 40%, which implies that, even at high SNR, the eavesdropper still cannot recover the data accurately.

For comparison, Figure 7 shows the security of the traditional CS. Similar to the encryption of STP-CS, only the original matrix can decrypt the data correctly, and the other three matrices cannot decrypt the data; the recovery errors of the other three matrices are larger than 80% from −5 dB to 30 dB. Based on this relative error, the performance of the traditional CS is better than STP-CS. But the dimension of the measurement matrix is 64×256, while the dimension of the measurement matrix is 2×8, so the security performance of STP-CS can be improved by increasing the size of the measurement matrix.

6. Conclusions

CS can fulfill the energy-efficiency and the encryption for big data transmission simultaneously. But the measurement matrix needs huge storage space, and the calculation cost of CS is large. In this paper, we propose STP-CS encryption to decrease the storage space for the secret key to save storage resource and reduce the calculation amount to save calculation resource. The simulation results show that the performance of saving resource is better compared with the traditional CS and CCS.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This paper is supported by the National Key Research and Development Program of China (Grants nos. 2016YFB0800602 and 2016YFB0800604), the National Natural Science Foundation of China (Grants nos. 61573067 and 61472045), the Beijing City Board of Education Science and Technology Project (Grant no. KM201510015009), and the Beijing City Board of Education Science and Technology Key Project (Grant no. KZ201510015015).

References

[1] Y. Lee, D. Blaauw, and D. Sylvester, "Ultralow power circuit design for wireless sensor nodes for structural health monitoring," *Proceedings of the IEEE*, vol. 104, no. 8, pp. 1529–1546, 2016.

[2] I. L. Santos, L. Pirmez, L. R. Carmo et al., "A decentralized damage detection system for wireless sensor and actuator networks," *IEEE Transactions on Computers*, vol. 65, no. 5, pp. 1363–1376, 2016.

[3] X. Ding, Y. Tian, and Y. Yu, "A real-time big data gathering algorithm based on indoor wireless sensor networks for risk analysis of industrial operations," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 3, pp. 1232–1242, 2016.

[4] L. Quan, S. Xiao, X. Xue, and C. Lu, "Neighbor-aided spatial-temporal compressive data gathering in wireless sensor networks," *IEEE Communications Letters*, vol. 20, no. 3, pp. 578–581, 2016.

[5] A. M. R. Dixon, E. G. Allstot, D. Gangopadhyay, and D. J. Allstot, "Compressed sensing system considerations for ECG and EMG wireless biosensors," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 6, no. 2, pp. 156–166, 2012.

[6] Y. Zhang, L. Y. Zhang, J. Zhou, L. Liu, F. Chen, and X. He, "A review of compressive sensing in information security field," *IEEE Access*, vol. 4, pp. 2507–2519, 2016.

[7] V. Abolghasemi, S. Ferdowsi, B. Makkiabadi, and S. Sanei, "On optimization of the measurement matrix for compressive sensing," in *Proceedings of the 18th European Signal Processing Conference*, pp. 427–431, August 2010.

[8] S. Sharma, A. Gupta, and V. Bhatia, "A new sparse signal-matched measurement matrix for compressive sensing in uwb communication," *IEEE Access*, vol. 4, pp. 5327–5342, 2016.

[9] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1982–2001, 2010.

[10] P. Song, J. F. C. Mota, N. Deligiannis, and M. R. D. Rodrigues, "Measurement matrix design for compressive sensing with side information at the encoder," in *Proceedings of the IEEE Statistical Signal Processing Workshop (SSP '16)*, pp. 1–5, IEEE, Palma de Mallorca, Spain, June 2016.

[11] R. R. Naidu, P. Jampana, and C. S. Sastry, "Deterministic compressed sensing matrices: construction via Euler squares and applications," *IEEE Transactions on Signal Processing*, vol. 64, no. 14, pp. 3566–3575, 2016.

[12] A. Ravelomanantsoa, H. Rabah, and A. Rouane, "Compressed sensing: a simple deterministic measurement matrix and a fast recovery algorithm," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 12, pp. 3405–3413, 2015.

[13] S. Li, F. Gao, G. Ge, and S. Zhang, "Deterministic construction of compressed sensing matrices via algebraic curves," *IEEE Transactions on Information Theory*, vol. 58, no. 8, pp. 5035–5041, 2012.

[14] J. Zhang, G. Han, and Y. Fang, "Deterministic construction of compressed sensing matrices from protograph LDPC codes," *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1960–1964, 2015.

[15] L. Yu, J. P. Barbot, G. Zheng, and H. Sun, "Compressive sensing with chaotic sequence," *IEEE Signal Processing Letters*, vol. 17, no. 8, pp. 731–734, 2010.

[16] G. Chen, D. Zhang, Q. Chen, and D. Zhou, "The characteristic of different chaotic sequences for compressive sensing," in *Proceedings of the 5th International Congress on Image and Signal Processing (CISP '12)*, pp. 1475–1479, IEEE, Chongqing, China, October 2012.

[17] D. Cheng and L. Zhang, "On semi-tensor product of matrices and its applications," *Acta Mathematicae Applicatae Sinica*, vol. 19, no. 2, pp. 219–228, 2003.

[18] D. Cheng, H. Qi, and A. Xue, "A survey on semi-tensor product of matrices," *Journal of Systems Science and Complexity*, vol. 20, no. 2, pp. 304–322, 2007.

[19] D. Cheng and Y. Dong, "Semi-tensor product of matrices and its some applications to physics," *Methods and Applications of Analysis*, vol. 10, no. 4, pp. 565–588, 2003.

[20] H. Yuan, H. Song, X. Sun, K. Guo, and Z. Ju, "Compressive sensing measurement matrix construction based on improved size compatible array LDPC code," *IET Image Processing*, vol. 9, no. 11, pp. 993–1001, 2015.

[21] W. U. Bajwa, J. D. Haupt, G. M. Raz, S. J. Wright, and R. D. Nowak, "Toeplitz-structured compressed sensing matrices," in *Proceedings of the IEEE/SP 14th Workshop on Statistical Signal Processing (SSP '07)*, pp. 294–298, IEEE, Madison, Wis, USA, August 2007.

[22] L. Applebaum, S. D. Howard, S. Searle, and R. Calderbank, "Chirp sensing codes: deterministic compressed sensing measurements for fast recovery," *Applied and Computational Harmonic Analysis*, vol. 26, no. 2, pp. 283–290, 2009.

[23] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: universal encoding strategies?" *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[24] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[25] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[26] T. T. Do, L. Gan, N. Nguyen, and T. D. Tran, "Sparsity adaptive matching pursuit algorithm for practical compressed sensing," in *Proceedings of the 42nd Asilomar Conference on Signals, Systems and Computers (ASILOMAR '08)*, pp. 581–587, IEEE, Pacific Grove, Calif, USA, October 2008.

[27] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 1094–1121, 2012.

[28] D. Xie, H. Peng, L. Li, and Y. Yang, "Semi-tensor compressed sensing," *Digital Signal Processing*, vol. 58, pp. 85–92, 2016.