A novel block encryption scheme based on chaos and an S-box for wireless sensor networks*

Tong Xiao-Jun(佟晓筠)[a)†], Wang Zhu(王 翥)[b)], and Zuo Ke(左 科)[a)]

[a)] *School of Computer Science and Technology, Harbin Institute of Technology, Weihai* 264209, *China*

[b)] *School of Information Technology and Engineering, Harbin Institute of Technology, Weihai* 264209, *China*

(Received 13 August 2011; revised manuscript received 5 September 2011)

The wireless sensor network (WSN) has been widely used in various fields, but it still remains in the preliminary discovery and research phase with a lack of various related mature technologies. Traditional encryption schemes are not suitable for wireless sensor networks due to intrinsic features of the nodes such as low energy, limited computation capability, and lack of storage resources. In this paper, we present a novel block encryption scheme based on the integer discretization of a chaotic map, the Feistel network structure, and an S-box. The novel scheme is fast, secure, has low resource consumption and is suitable for wireless sensor network node encryption schemes. The experimental tests are carried out with detailed analysis, showing that the novel block algorithm has a large key space, very good diffusion and disruptive performances, a strict avalanche effect, excellent statistical balance, and fast encryption speed. These features enable the encryption scheme to pass the SP800-22 test. Meanwhile, the analysis and the testing of speed, time, and storage space on the simulator platform show that this new encryption scheme is well able to hide data information in wireless sensor networks.

Keywords: wireless sensor networks, chaos encryption, discretization of chaotic equations, S-box

PACS: 05.45.Gg **DOI:** 10.1088/1674-1056/21/2/020506

1. Introduction

The wireless sensor network (WSN) is composed of a set of ad-hoc sensors, which aim to collaborate perception, acquisition, and processing of information of objects perceived in the network coverage geographic area.[1] If WSNs are to be used in fields sensitive to the information, security will become the primary problem. If important information is leaked or tampered with, the loss will be enormous, ranging from individual life in a battlefield situation and even to national security. So, the security of WSNs is dependent on whether or not to spread important information in the world. After a long period of accumulation and development, conventional encryption technologies (such as, Data Encryption Standard (DES), Advanced Encryption Standard (AES), RSA, etc.) can security protect information very well. However, the WSN is a new network and many key WSN technologies at still at an early stage. Few encryption schemes are suited to WSNs due to some intrinsic features of sensor nodes such as low energy, limited computation capability, and storage resources, so the traditional encryption schemes, such as DES, AES, and RSA, are not suitable for WSNs due to the demand for more hardware resources. At the same time, research on the existing security mechanism for WSNs is inadequate, and current research on WSNs focuses mainly on hardware, communications routing protocols, etc., and the research on data encryption security is still in its infancy at home and abroad, so it is very necessary to research new security theories and encryption technologies for WSNs.[2−4] With the development of encryption technology, chaos encryption is beginning a develop well, as well as research on aspects of randomness, unpredictability of its motion states, and sensitive dependence on initial parameters. These features of chaos meet the information security requirements of cryptography, so there have been many chaos en-

*Project supported by the National Natural Science Foundation of China (Grant No. 60973162), the Natural Science Foundation of Shandong Province, China (Grant No. ZR2009GM037), the Science and Technology Fund of Shandong Province, China (Grant No. 2010GGX10132), the Scientific Research Foundation of Harbin Institute of Technology at Weihai, China (Grant No. HIT(WH) ZB200909), the Key Natural Science Foundation of Shandong Province, China (Grant No. Z2006G01), the Technology Research and Development Program of Weihai High-Technology Development Zone in Shandong Province, China (Grant No. 201025), and the Technology Research and Development Program of Weihai, China (Grant No. 2008011).

†Corresponding author. E-mail: tong_xiaojun@163.com

© 2012 Chinese Physical Society and IOP Publishing Ltd

cryption schemes proposed in recent decades,[5−8] especially in chaos image encryption. The outstanding performance of chaos in the traditional encryption areas and its good nonlinear characteristic slowly make the chaos begin to be applied to the WSNs. At the same time, with the interaction between and the development of chaos and traditional cryptography, chaos gradually has been widely applied to encryption fields. Chaos shows a new direction of its application in WSNs.[9]

Seemingly, chaos is irregular, similar to a random phenomenon in a deterministic nonlinear system, and is a complicated form of motion ubiquitously existing in nature. Instead of a simple disorder, chaos is rich in internal levels of ordered structure with no obvious cycle and symmetry.[10] Chaos is widely used in the fields of cryptography and communication security due to its good features such as mixing chaotic orbits, randomness, and sensitivity to system parameters.[11−14] Applying chaos to WSNs is a new direction for the research of encryption schemes in WSNs due to the lack of existing security mechanisms and great potential application in the WSN field. The author of Ref. [15] proposed a kind of integer chaotic encryption scheme (LCS) using limited resources and energy wireless sensor nodes with a logistic map in WSNs. However, the author of Ref. [16] has discussed and analysed the encryption scheme, pointing out that the encryption scheme with its intrinsic imperfectness and low security in Ref. [15] can be deciphered using differential analyses. According to the characteristics of a WSN sensor and the situation mentioned above, in the present paper, we propose a novel block encryption scheme based on chaos and S-box technology (CSS) in order to meet the resource requirements and to enhance the data encryption security for WSN nodes.

The rest of the present paper is organized as follows. In Section 2, we describe some general traditional block encryption schemes for WSNs, including RC5, RC6, AES-Rijindael, and SKIPJACK. In Section 3, we propose and discuss the discretized chaos equation and construction of an S-box for WSNs. In Section 4, we design and implement a new encryption scheme based on chaos and an S-box, mainly including the outline of block encryption and decryption for this encryption scheme. In Section 5, the security of the new encryption scheme is evaluated via both cryptanalysis and experiments. Finally, in Section 6, we draw some conclusions from the present study.

2. Traditional encryption scheme for WSNs

2.1. RC5 encryption scheme

RC5 is a block cipher with variable parameters: block size (32 bits, 64 bits, and 128 bits), key size and encryption rounds, and it can be expressed as RC5—w/r/b. It was designed by Ron Rivest and analysed by the RSA laboratory.[17,18] RC5 algorithm is a very compact algorithm with its data processing using only the general operation for the common microprocessors, such as modular addition, XOR, and cyclic shift. In RC5 two w-bits registers are used to store the plaintext input, and the ciphertext is stored in the same registers. Because it is a symmetric encryption algorithm, the decryption is the inverse of encryption. From what this algorithm shows us, RC5 makes the input data depend on the number of cyclic shifts in order to achieve the capacity that the number of the cyclic shifts cannot be predicted. With the features of low storage space, fast speed, and variable number of rounds and key length, RC5 is widely used in WSNs. With in-depth theoretical analysis, we find that RC5 has some security risks because of its intrinsic weak diffusion.[19] Concrete algorithm can be found in Ref. [20].

2.2. RC6 encryption scheme

RC6 is a new block cipher submitted to NIST for consideration as the new Advanced Encryption Standard (AES). The design of RC6 began with a consideration of RC5 as a potential candidate for an AES submission. Modifications were then made to meet the AES requirements, to increase security, and to improve performance.[21] The inner loop, however, is based on the same "half-round" found in RC5. RC6 is a further development based on RC5 by using the quadratic function.

Function $f(x) = x(2x+1)$ is used to enhance the diffusion rate, so RC6 can increase the security with less loops than RC5. At the same time, RC6 handles 128-bits input/output blocks with the variable key size and the number of rounds, so users can flexibly set the parameter of RC6 to meet the future growth and market demands. Concrete algorithm is seen in Ref. [22].

2.3. AES-Rijndael encryption scheme

AES-Rijndael is an iterative block cipher and designed by Joan Daeman and Vincent Rijmen in Bel-

gium in response to the AES. It is composed mainly of nonlinear components, linear components, and round keys, and though it employs an iterative structure, is does not have a Feistel network structure but an SP structure instead. The AES specifies the Rijndael algorithm, a symmetric block cipher that can process data blocks of 128 bits, using cipher keys with lengths of 128, 192, and 256 bits.[23] The AES-Rijndael encryption scheme was designed to handle additional block sizes and key lengths. However they are not adopted in this standard. AES uses the SP (substitute permutation) network structure with the 128-bits blocks and three optional key lengths of 128-bits, 192-bits, and 256-bits. Round r depends on the key length. If key length is 128-bits, the $r = 10$; if the key length is 192-bits, the $r = 12$; if the key length is 256-bits, the $r = 14$.[24] Concrete algorithm is seen in Ref. [25].

2.4. Skipjack encryption scheme

The Skipjack encryption algorithm is also a block cipher and has a key length of 80-bits, plaintext, a ciphertext length of 64-bits, and 32 rounds, and its input sequence is the same as the sequences of encryption and decryption.

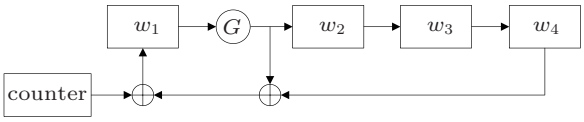


Fig. 1. Rule A.

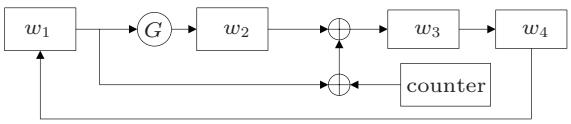


Fig. 2. Rule B.

In the encryption process A and B rules (see Figs. 1 and 2) are used. The input plaintext w_i^0, $0 \le i \le 4$, the encryption algorithm is 16 bits. The start counter is set to be 1, rule A implements 8 rounds, and then the rule B implements 8 rounds, and then rule A implements 8 rounds again, and rule B implements 8 rounds again, finally the ciphertext is outputted: w_i^{32}, $0 \le i \le 4$. In the decryption process, A^{-1} and B^{-1} rules with the process opposite to the encryption process are used. Dependent variable of 16-bits key space displacement G has a Feistel structure with four rounds, the round function F is a fixed byte substitution table, the table requires $16 \times 16 = 256$ bytes of storage units, each round of G contains one byte key. The concrete algorithm can be found in Ref. [26].

3. Discretized chaos equation and S-box construction for WSNs

The chaos tent map is a typical chaotic system for the real field. However, the embedded system for the wireless micro-sensor network node is not good at handling floating-point, multi-byte division, and other operations, so the study of the discrete chaotic equation will help the WSN node process the chaotic operations.

The S-box is a key component for the design of symmetric key block encryption algorithm because it plays a unique nonlinear transformation role in the block encryption algorithm. A good S-box can enhance the security of its encryption system against a variety of exterior attacks such as statistical attacks and differential attacks. In the following, we will describe the tent map and its discretization method, and introduce a construction method of 4×4 S-box according to the orthomorphic permutation which will be used to construct a new 4×4 S-box for WSNS.

3.1. Tent map and its discretization

Generally the tent map is defined as follows:

$$x_{n+1} = \begin{cases} \dfrac{x_n}{a}, & 0.5 \le x_n < 1, \\ \dfrac{(1-x_n)}{(1-a)}, & 0 \le x_n < 0.5. \end{cases} \tag{1}$$

The standard tent map is obtained with $a = 0.5$. The so-called oblique tent map is obtained with $a \neq 0.5$. By further extension, a class of piecewise linear maps can be obtained as follows:

$$x_{n+1} = \begin{cases} \dfrac{x_n}{p}, & 0 \le x_n < p, \\ \dfrac{(x_n - p)}{(0.5-p)}, & p \le x_n < 0.5, \\ \dfrac{(1-x_n-p)}{(0.5-p)}, & 0.5 \le x_n < 1-p, \\ \dfrac{(1-x_n)}{p}, & 1-p \le x_n \le 1. \end{cases} \tag{2}$$

The outstanding feature of tent map is its distribution function with uniformity, and its behaviour is chaotic with the elongation and folding properties in the real field. If we equivalently transform the tent

map from the real field to the integer field, making it an integer tent map, the features of elongation and folding will be kept.

Multiplying both sides of formula (2) by a, supposing $p = 1/4$, $z_n = ax_n$, and then $z_{n+1} = ax_{n+1}$, substituting them into formula (2) and simplifying the resulting formula, we can obtain

$$Z_{n+1} = \begin{cases} 4Z_n, & 0 \le Z_n < \dfrac{1}{4}a, \\ 4Z_n - a, & \dfrac{1}{4}a \le Z_n < \dfrac{1}{2}a, \\ 3a - 4Z_n, & \dfrac{1}{2}a \le Z_n < \dfrac{3}{4}a, \\ 4a - 4Z_n, & \dfrac{3}{4}a \le Z_n \le a. \end{cases} \tag{3}$$

Take $a = 2^W$, where W is the computer word size, and take only the integer values of formula (3), then $Z_n \in [0, 2^W]$ and formula (3) acts as an unsigned integer iteration operation within the range $[0, 2^W]$. For formula (3), we can quickly complete the integer arithmetic only through shift, multiplication, and addition (complement) operations. Therefore, the chaotic iteration formula (3) is well suited to the WSN node processing the integer arithmetic on the embedded systems.

We can obtain two stable zero solutions: 0 and a for all the integer solutions of formula (3). In a limited discrete binary operation, even if the initial iteration value is not zero, we will obtain the value a from formula (3) after performing several iterations on formula (3) and when the iteration value is a, the iteration result of formula (3) will maintain value 0 no matter how many iterations have been performed. In order to eliminate this stable phenomenon, a new formula is proposed on the base of formula (3) as follows:

$$Z_{n+1} = \begin{cases} 4Z_n + 1, & 0 \le Z_n < \dfrac{1}{4}a, Z_n \text{ is even}, \\ 4Z_n - 1, & 0 \le Z_n < \dfrac{1}{4}a, Z_n \text{ is odd}, \\ 4Z_n - a, & \dfrac{1}{4}a \le Z_n < \dfrac{1}{2}a, \\ 3a - 4Z_n - 1, & \dfrac{1}{2}a \le Z_n < \dfrac{3}{4}a, \\ 4a - 4Z_n + 1, & \dfrac{3}{4}a \le Z_n < a, Z_n \text{ is odd}, \\ 4a - 4Z_n - 1, & \dfrac{3}{4}a \le Z_n < a, Z_n \text{ is even}. \end{cases} \tag{4}$$

We can obtain $4a - 4Z_n + 1 < a$, $4a - 4Z_n + 1 \in [5, a-3]$, $a = 2^W$ from formula (4). Similarly, we can see that the iteration values of formula (4) will not induce a stable zero solution within $[0, a-1]$. In this way, the improved formula (4) well eliminates the specific cases for the iteration values of -1 and greater than or equal to a by controlling the parity of Z_n. For the piecewise linear chaotic map with the stretching and folding features, formula (4) satisfies the closed theorem within $[0, a-1]$.

Therefore, we can conclude that the improved integer formula (4) based on the piecewise linear chaotic map is an integer algebra system within $[0, a-1]$.

3.2. Construction of 4×4 S-boxes based on orthomorphic permutation

Suppose that $\theta : x \to \theta(x)$ is a bijection based on $GF(2^n)$ and if $\phi : x \to \phi(x) = x \oplus \theta(x)$ is also a bijection, then θ is defined as orthomorphic permutation.

Orthomorphic permutation can be equivalent to a sequence that can be found from the orthogonal Latin square table (seeing Table 1) by satisfying the demand that each row or column is just to take a number from Table 1, and this sequence must be a replacement based on $F_2^n \to F_2^n$.

Table 1. Orthomorphic permutation table.

0	1	2	$\cdots$	$2^n - 1$
$1 \oplus 0$	$1 \oplus 1$	$1 \oplus 2$	$\cdots$	$1 \oplus 2^n - 1$
$2 \oplus 0$	$2 \oplus 1$	$2 \oplus 2$	$\cdots$	$2 \oplus 2^n - 1$
$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
$2^n - 1 \oplus 0$	$2^n - 1 \oplus 1$	$2^n - 1 \oplus 2$	$\cdots$	$2^n - 1 \oplus 2^n - 1$

According to the orthogonal Latin square table (see Table 1), we can obtain an orthogonal Latin square result of the operation table (see Table 2 below). Constructing an orthomorphic permutation from Table 2 must satisfy the following rules:

(i) each cell in Table 2 can only take one number;

(ii) each row and column should respectively take two cells, the same cell must be taken twice;

(iii) the first and fourth part should take a total of eight cells, the second and third part should also take a total of eight cells;

(iv) Table 2 shows the four parts, each part respectively takes four cells.

If you traverse through Qiongsou method in Table 2 for all possible orthomorphic permutations and you obtain $16! = 20922789888000$ possibilities of the 4×4 S-box, which in practical application is impossible. So the author presents the general steps to generate a series of 4×4 S-box in Ref. [27] as follows.

i) As the structure of four parts in Table 2 is the same, according to rules (ii) and (iv) given above, all

the possible results of one part are stored in an array M.

ii) According to the above rule (ii), construct three 1232×1232 arrays A, B, C, in which $A[i][j]$ indicates that the first part result is $M[i]$ and the fourth part result is $M[j]$. If the results meet the above rule (ii) and are marked as true, otherwise false, $B[i][j]$ indicates that the first part result is $M[i]$ and the second part result is $M[j]$. If the results meet the above rule (ii) and are marked as true, otherwise false, $C[i][j]$ indicates that the first part result is $M[i]$ and the third part result is $M[j]$. If the results meet the above rule (ii) and are marked as true, otherwise false, the relationship between the fourth part, the second part, and the third part is the same as the relationship between the first part, the second part, and the third part.

iii) Through all the possible result combinations of the four parts, the legitimate result combinations will be stored.

Table 2. Orthogonal Latin square result of the operation table.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
	1	0	3	2	5	4	7	6	9	8	11	10	13	12	15	14	
	2	3	0	1	6	7	4	5	10	11	8	9	14	15	12	13	
	3	2	1	0	7	6	5	4	11	10	9	8	15	14	13	12	
I	4	5	6	7	0	1	2	3	12	13	14	15	8	9	10	11	II
	5	4	7	6	1	0	3	2	13	12	15	14	9	8	11	10	
	6	7	4	5	2	3	0	1	14	15	12	13	10	11	8	9	
	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	
	8	9	10	11	12	13	14	15	0	1	2	3	4	5	6	7	
	9	8	11	10	13	12	15	14	1	0	3	2	5	4	7	6	
	10	11	8	9	14	15	12	13	2	3	0	1	6	7	4	5	
III	11	10	9	8	15	14	13	12	3	2	1	0	7	6	5	4	IV
	12	13	14	15	8	9	10	11	4	5	6	7	0	1	2	3	
	13	12	15	14	9	8	11	10	5	4	7	6	1	0	3	2	
	14	15	12	13	10	10	8	9	6	7	4	5	2	3	0	1	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	

We can find the legitimate replacement from the stored combination of step III, in which only one number is taken from each row and column in Table 2, respectively and each number in Table 2 is taken only once. Through the above steps, we can find many 4×4 S-boxes that meet the requirements of orthomorphic permutations, and these 4×4 S-boxes generated by the orthomorphic permutation all can meet the demands for the nonlinearity and differential uniformity of S-box.[27]

Table 3 is a 4×4 S-box generated by the orthomorphic permutation, and its nonlinearity and differential uniformity all meet the demands for the S-box. Applying the S-box technology to the design of the encryption algorithm is considered to improve the storage space and the security. As is well known, the general S-box needs a relatively large storage space, but the 4×4 S-box based on the orthomorphic permutation only needs 8-bytes storage space, so it is worth using small space in exchange for high security.

Table 3. 4×4 S-box substitution values for the byte xy (in quaternary format).

x \ y	0	1	2	3
0	13	20	03	30
1	12	21	02	31
2	22	00	33	11
3	23	10	32	01

4. Encryption scheme based on chaos and S-box for WSNs

As is well known, there are several types of network structures used in the design of an encryption scheme such as Feistel network (DES, FEAL, TWOFISH, LOKl97, GOST), variant Feistel network (RC5, MISTY2, CAST-256), and the SP network (such as IDEA, Rijndael, SAFER).[28] In this paper we propose a block encryption scheme based on the discretized chaotic map (piecewise linear maps) and S-box technology using the Feistel network structure with a block length of 32 bits, a key length of 128 bits, CBC mode, an initial vector of 32 bits and 14 rounds. In order to meet the node's hardware requirements, this encryption scheme of data processing is based on 8 bits as a unit for the nodes' CPU word length of 8 bits. Used in this encryption scheme are only shift, simple multiplication, subtraction (complement), and addition operations that are very suited to the embedded system software processing on the nodes of WSNs.

4.1. Research of encryption scheme based on chaos and an S-box

In this paper, we use the encryption scheme based on the Feistel structure that a round function F acts on the plaintext block several times. Supposing that r rounds of the Feistel structure encryption scheme are performed and the block length is $2n$ bits, then each round of operation can be formally defined as follows:

$$\text{Round}_i : L_{i-1}||R_{i-1} \mapsto R_i||F(K_i, R_{i-1}) \oplus L_{i-1}, \quad (5)$$

where $i = 1, 2, \ldots, r$, L_i and R_i denote the left part and the right part of the i-th block with a length of n bits, and K_i is the i-th round sub-key generated by the master key through a certain algorithm. The nonlinear function F is the core component of the encryption scheme and can cause confusion. Figure 3 shows the overall description of encryption scheme based on chaos and S-box. This scheme can handle 32 bits once with directly calling the basic single-byte operation instructions built in CPU and the process is less time-consuming, fast speed, and low energy consumption for the basic single-byte instructions of 8 bits integer operation. There are 14 rounds for the implementation of the encryption scheme with each round using a sub-key, and each sub-key also contains four single-byte components ($K_i(1)$, $K_i(2)$, $K_i(3)$, $K_i(4)$). These sub-keys are generated through a certain algorithm. At the same time, in order to enhance the security of this algorithm, the round function F is designed by the chaos and the S-box technology. Except the first round and the last round, the remaining rounds all perform the same operations. In order to enhance the diffusion of this algorithm, we separately place a substitution function for handling 32 bits once at the beginning and the end of this algorithm.

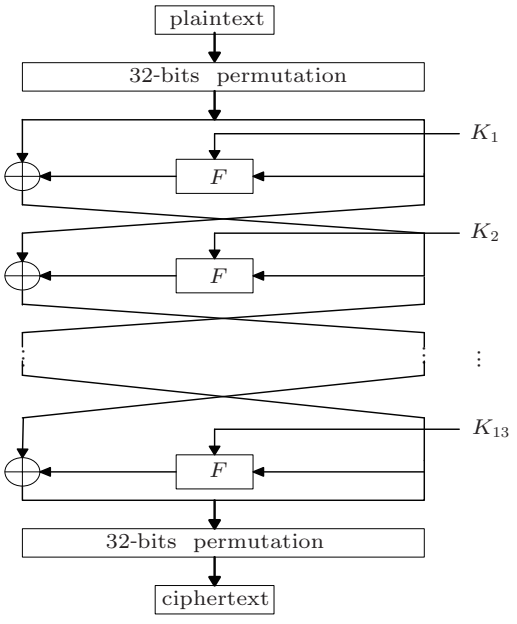


Fig. 3. Encryption model based on Feistel structure with chaos and S-box.

4.2. Research on the construction of F

Figure 4 shows the internal structure of F, where $\oplus$ is the XOR operation, $\boxplus$ is the addition operation for module 2^8, S-box is a 4×4 S-box (see Table 3) based on orthomorphic permutation, f is the discretized piecewise linear map formula (4), $K_r(t)$ ($t = 1, 2, 3, 4$) is the four single-bye components for the r-th sub-key. The F is dependent mainly on the implementation of S-box and function f, and the inputs of S-box depend on $K_r(1)$, $K_r(3)$, and the outputs of the last round, and the inputs of f depend on not only the outputs of last round but also the outputs of S-box, which aims to enhance diffusion and confusion effects.

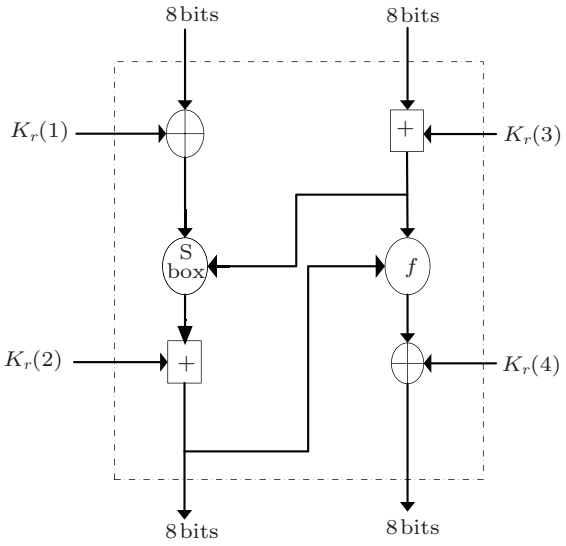


Fig. 4. Internal structure for F.

Under the control of the sub-key $K_r(t)$, ($t = 1, 2, 3, 4$), F function will transform the two input sub-blocks of 8-bits into the two 8-bits output sub-block by the action of XOR, addition for module 2^8, the S-box and f. The i-th sub-key K_i is generated by the algorithm of Subsection 4.3 with the master key $K = K_1 K_1, \ldots, K_{16}$.

4.3. Algorithm for the key expansion

The key is an important factor for the design of the encryption algorithm. Considering the security of the encryption algorithm, we should establish the relationship between the complex of the sub-keys and the master key.

As is well known, Rivest[20] designed the RC5 key expansion algorithm with the features of simple operations and high security in 1994. Based on the above fact, the key expansion algorithm in this paper is designed by modifying the RC5 key expansion in order to meet the requirements for the nodes in WSNs. Specific schemes are shown as follows:

(I) Initializing the key vector S

First, we can obtain the value of P_8, Q_8 by the P_w, Q_w formula ($w = 8$) for RC5,[29,30] and then initialize the key vector S by the following pseudo-code:

$S[0] = P_8$;
for ($i = 1$; $i < t - 1$; $i++$)
$S[i] = S[i-1] + Q_8$;

where $t = 16$.

As each round operation will use a sub-key containing 4 single-bye components, the implementation for 14 round iterations will observe the space of 56-bytes. Although the longer the key length, the higher the security is, the node in the WSNs has the limited space. So in this paper a compromise approach suited for the WSNs is adopted. This approach can meet the requirements of security and space for the nodes by the transformation formula $S[i] \rightarrow S[i\%16]$, $(i = 0, 1, \ldots, 55)$.

(II) Generating the round keys

We can copy the key $K[0 \cdots 15]$ of 16 bytes into the vector $L[0 \cdots 15]$, and then obtain encryption key vector S by the following pseudo-code:

$A = B = 0$; $i = j = 0$; $k = 3 * t$;
for (; $k > 0$; k- -) {
$A = \mathrm{ROTL}(S[i] + A + B, 3)$; $S[i] = A$;
$B = \mathrm{ROTL}(L[j] + A + B, A + B)$; $L[j] = B$;
$i = (i + 1)\%t$; $j = (j + 1)\%t$; },

where $t = 16$, the ROTL(X, n) is the left cyclic shift n for X.

4.4. Algorithm of the decryption process

The Feistel network structure has the good feature of having the same structure for the encryption and decryption, so the decryption is just the inverse operation of the encryption with the reverse order of the sub-key; and then the function F can be designed to be very complicated and cannot require it to be reversible, since the feature of Feistel network structure can easily ensure the block encryption algorithm no matter whether the F is reversible. These characteristics can clearly be seen from the following formula:

$$\begin{aligned} & R_i \oplus F(L_i, K_i) \\ = & (L_{i-1} \oplus F(R_{i-1}, K_i)) \oplus F(L_i = R_{i-1}, K_i) = L_{i-1}. \end{aligned}$$

5. Experimental tests and security analysis

5.1. Tests for diffusion and confusion

Shannon in his classic paper, Ref. [31], proposed two basic design principles for the block encryption, namely, diffusion and confusion. The degrees of diffusion and confusion for the block cipher algorithm can be measured by the statistical test of nonlinear diffusion. The analysis of nonlinear diffusion for the encryption algorithm usually refers to the completeness, the avalanche, and the strict avalanche criterion. The completeness and the avalanche were first introduced by Kam Davida and Feistel, while Webster and Tavares have further proposed the concept of strict avalanche criterion.

For a vector $x = (x_1, \ldots, x_n) \in (GF(2))^n$, the vector $x^{(i)} \in (GF(2))^n$ denotes the vector obtained by complementing the i-th bit of x (for $i = 1, \ldots, n$). The Hamming weight $w(x)$ of x is defined as the number of nonzero components of x.

The dependence matrix of a function $f : (GF(2))^n \rightarrow (GF(2))^m$ is an $n \times m$ matrix A, whose (i, j)-th element a_{ij} denotes the number of inputs for which complementing the i-th input bit results in a change of the j-th output bit, i.e.

$$a_{ij} = \#\{x \in (GF(2))^n | (f(x^{(i)}))_j \neq (f(x))_j\}$$

for $i = 1, \ldots, n$ and $j = 1, \ldots, m$.

The distance matrix of a function $f : (GF(2))^n \rightarrow (GF(2))^m$ is an $n \times (m+1)$ matrix B, whose (i, j)-th element b_{ij} denotes the number of inputs for which complementing the i-th input bit results in a change of the j output bit, i.e.

$$b_{ij} = \#\{x \in (GF(2))^n | w(f(x^{(i)}) - f(x)) = j\}$$

for $i = 1, \ldots, n$ and $j = 0, \ldots, m$.

Of course, unless the number of input bits is small, it is impossible to compute the dependence and distance matrices for all possible inputs. Therefore, one usually considers a "suitable" number of randomly chosen inputs. The dependence and distance matrices are then defined as follows:

$$a_{ij} = \#\{x \in X | (f(x^{(i)}))_j \neq (f(x))_j\}$$

for $i = 1, \ldots, n$ and $j = 1, \ldots, m$;

$$b_{ij} = \#\{x \in X | w(f(x^{(i)}) - f(x)) = j\}$$

for $i = 1, \ldots, n$ and $j = 0, \ldots, m$, where X is a "suitable" randomly chosen subset of $(GF(2))^n$.

Now assume that we have computed the dependence matrix A and the distance matrix B of a function $f : (GF(2))^n \rightarrow (GF(2))^m$ for a set X of inputs, where X is either $(GF(2))^n$ or a random subset of

$(GF(2))^n$.[28] The degree of completeness of f is defined as

$$d_1 = 1 - \frac{\#\{(i,j)|a_{ij} = 0\}}{nm}.$$

The degree of avalanche effect of f is as follow:

$$d_2 = 1 - \frac{\sum_{i=1}^{n} \left| \frac{1}{\#X} \sum_{j=1}^{m} 2jb_{ij} - m \right|}{nm}$$

or

$$d_3 = \frac{\sum_{i=1}^{n} \left(\sum_{x \in X} w(f(x^{(i)}) - f(x)) \right)}{\#X \times nm}.$$

The degree of strict avalanche criterion of f is defined as follows:

$$d_4 = 1 - \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} \left| \frac{1}{\#X} 2a_{ij} - 1 \right|}{nm}.$$

For the function f to have good degrees of completeness, avalanche effect, and strict avalanche criterion, numbers d_1, d_2, d_3, d_4 must satisfy $d_1 = 1$, $d_2 \approx 1$, $d_3 \approx 0.5$, $d_4 \approx 1$.[32]

In this paper we will test the nonlinear diffusion for the six encryption algorithms which are RC5, RC6, CSS, AES-Rijndael, SKIPJACK, and LCS. In the way, LCS algorithm only adopts a single logistic chaotic mapping to achieve the encryption, while CSS adopts the mixing technology of chaos and S-box. In addition to the above difference, the rest for LCS and CSS are the same. The 3.6×10^5 random 32 bits unsigned integer for the test input vector is generated by Matlab.

Table 4 shows the results of the nonlinear diffusion for the plaintext to ciphertext. And Table 5 shows the results of the nonlinear diffusion for the master-key to ciphertext, where the test parameters for RC5 are the input/output of 64 bits, the key length of 256 bits, and the iteration rounds of 20 (recommended value), and the test parameters for RC6 are the input/output of 128 bits, the key length of 256 bits, and the iteration rounds of 20 (recommended value). The test parameters for CSS or LCS are the input/output of 32 bits, the key length of 128 bits, and the iteration rounds of 14 or 13 respectively; the test parameters for AES-Rijndael are the input/output of 128 bits, the key length of 256 bits, and the iteration rounds of 14 (maximum value). The test parameters for Skipjack are the input/output of 64 bits, the key length of 80 bits, and the iteration rounds of 32 (recommended value).

From Tables 4 and 5, we can see that LCS does not satisfy the basic requirements of nonlinear diffusion. While the CSS can meet the basic requirements with the good completeness and avalanche effect; and the metric of the nonlinear diffusion for the CSS is generally better than other algorithms (especially than the LCS). So we can conclude that the CSS can satisfy the strict avalanche criterion and resist the differential cryptanalysis with the good features of diffusion and confusion. This shows the combination between traditional S-box and chaos technology has more advantages in security than the single logistic chaos. This is why we use the CSS instead of the LCS for the encryption.

Table 4. Results of the nonlinear diffusion for plaintext to ciphertext.

Algorithm	Metric of the nonlinear diffusion (d_1, d_2, d_3, d_4)			
RC5	d_1=1.000000	d_2=0.500018	d_3=0.999786	d_4=0.999786
RC6	d_1=1.000000	d_2=0.499991	d_3=0. 999779	d_4=0.999779
CSS	d_1=1.000000	d_2=0.500005	d_3=0.999811	d_4=0.999811
AES-Rijndael	d_1=1.000000	d_2=0.499977	d_3=0.999781	d_4=0.999781
Skipjack	d_1=1.000000	d_2=0.499975	d_3=0.999761	d_4=0.999761
LCS	d_1=0.939453	d_2=0.470363	d_3=0.940405	d_4=0.940405

Table 5. Results of the nonlinear diffusion for the master-key to ciphertext.

Algorithm	Metric of the nonlinear diffusion (d_1, d_2, d_3, d_4)			
RC5	d_1=1.000000	d_2=0.500003	d_3=0.999770	d_4=0.999770
RC6	d_1=1.000000	d_2=0.500004	d_3=0. 999799	d_4=0.999799
CSS	d_1=1.000000	d_2=0.500008	d_3=0.999796	d_4=0.999796
AES-Rijndael	d_1=1.000000	d_2=0.499997	d_3=0.999763	d_4=0.999763
Skipjack	d_1=1.000000	d_2=0.500013	d_3=0.999765	d_4=0.999765
LCS	d_1=1.000000	d_2=0.499910	d_3=0.999626	d_4=0.999626

5.2. Uniform distribution analysis for ASCII values and 0–1 binary sequence

In order to verify the performance of the uniform distribution for the CSS algorithm, we implement the test of uniform distribution for ASCII values and 0–1 binary sequence. In this paper, we select a 14.6-MB text file to be encrypted (CBC encryption mode) for the test with a random initial key of $16B$, an initial vector of $4B$ and encryption rounds of 14.

From Figs. 5 and 6 we can clearly see that the distributions of ASCII values for the plaintext and ciphertext are very different. In theory, if a certain encryption algorithm has a good uniform distribution in order to hide the non-uniform distribution of the plaintext, the proportion of each ASCII character in the ciphertext is about 0.00390625. From Fig. 6, we can obviously obtain the result that the CSS algorithm observes the good uniform distribution of ASCII values. Figure 7 shows the distribution of 0–1 binary sequence for the ciphertext of these six block encryption algorithms by the statistical analysis method. From Fig. 7, we can obtain the information that the proportions of 0–1 binary sequence for the six algorithms are all close to 1 with the increase of the test sequence length, but the balance fluctuation of the CSS algorithm is smaller than those of the others.

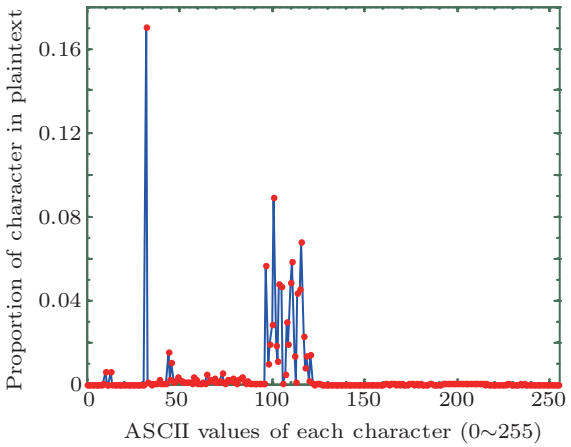


Fig. 5. (colour online) Distribution of ASCII values for the plaintext.

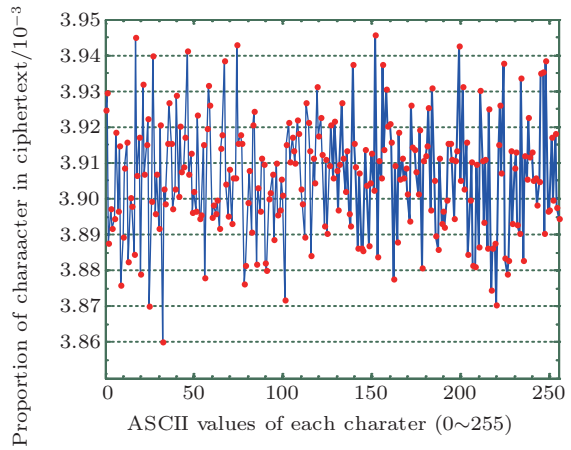


Fig. 6. (colour online) Distribution of ASCII values for the ciphertext.

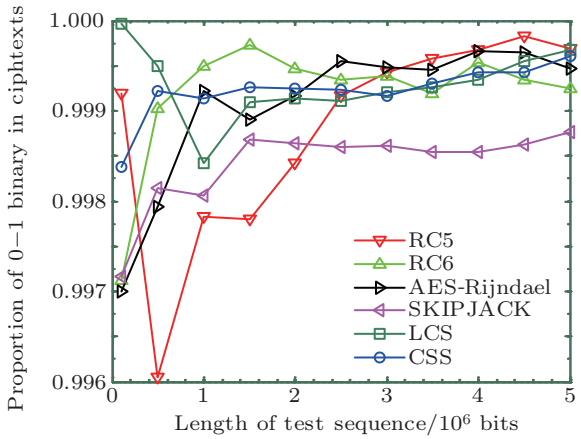


Fig. 7. (colour online) Distributions of 0–1 binary sequence for the ciphertexts.

5.3. Entropy test

Entropy formula can be formally defined as follows:

$$H(S) = \sum_{S}^{P(s_i)} \log_2 \frac{1}{P(s_i)},$$

where $P(s_i)$ denotes the probability of each symbol appearance. A statistical unit is 8 bytes. If the probability of every symbol in accordance with uniform distribution would be 1/8, in theory, the entropy should be 8. Therefore, the entropy of a good encryption algorithm should be as close to the value of 8 as possible. Table 6 shows the entropy values of the six encryption algorithms. From Table 6, we can see the entropy value of CSS is close to 8 higher than those of other four algorithms, which shows that the CSS has a very good entropy test performance.

Table 6. Entropy values for the six encryption algorithms.

Entropy	RC5	RC6	CSS	AES-Rijndael	Skipjack	LCS
$H(S)$	7.999982	7.999989	7.999991	7.999987	7.999991	7.999989

5.4. Key space analysis

A good encryption algorithm must not only have the sensitivity of the keys (as shown in Table 5), but also require the internal details of the encryption algorithm which can be made public. Its security depends entirely on the key security, so the key space should be large enough to be able to resist brute-force attacks.

In this paper, the CSS encryption algorithm has a key length of 128 bits with using the CBC encryption mode, and the 32 bits initial vector can be used as a secondary key, so the key space of the CSS can be up to $2^{160} \approx 1.46 \times 10^{48}$. Therefore, under the current computing power, the successful cryptanalysis to the CSS is temporarily unavailable with such types of attack as brute-force attacks on key, dictionary attacks and key matching. At the same time, the CSS has a good key expansion algorithm, which does not make the obviously weak key and semi-weak key appear in the key space with the enhancement of the security of the algorithm to a certain extent.

5.5. SP 800-22 test

During the test for the SP800-22, 122774400 bits are divided into 100 groups that are analysed. Tables 7 and 8 respectively show the final test results for CSS and LCS.

Table 7. SP 800-22 test results for CSS.

Statistical test		Proportion	p-value	Results
Frequency		99/100	0.334538	success
Block frequency ($m = 12280$)		100/100	0.224821	success
Cumulative sums	forward	99/100	0.304126	success
	reverse	99/100	0.236810	success
Runs		98/100	0.759756	success
Longest run ($M = 10000$, $N = 100$)		99/100	0.699313	success
Rank		100/100	0.779188	success
FFT		100/100	0.153763	success
Nonoverlapping template ($m = 9$, $B = 000010011$)		100/100	0.924706	success
Overlapping template ($m = 9$, $M = 1032$, $N = 968$)		100/100	0.383827	success
Universal ($L = 7$, $Q = 1280$, $K = 141577$)		100/100	0.350485	success
Approximate entropy ($m = 10$)		96/100	0.075719	success
Random excursions ($x = -3$)		67/69	0.551026	success
Random excursions variant ($x = 5$)		69/69	0.452799	success
Serial ($m = 16$)	P-value1	99/100	0.955835	success
	P-value2	100/100	0.275709	success
Linear complexity ($M = 1000$)		100/100	0.595549	success

Table 8. SP800-22 test results for LCS.

Statistical test		Proportion	p-value	Results
Frequency		99/100	0.304126	success
Block frequency ($m = 12280$)		99/100	0.911413	success
Cumulative sums	forward	99/100	0.924076	success
	reverse	99/100	0.595549	success
Runs		98/100	0.657933	success
Longest run ($M = 10000$, $N = 100$)		99/100	0.262249	success
Rank		99/100	0.534146	success
FFT		96/100	0.162606	success
Nonoverlapping template ($m = 9$, $B = 000010011$)		99/100	0.779188	success
Overlapping template ($m = 9$, $M = 1032$, $N = 968$)		100/100	0.191687	success
Universal ($L = 7$, $Q = 1280$, $K = 141577$)		97/100	0.759756	success
Approximate entropy ($m = 10$)		100/100	0.153763	success
Random excursions ($x = -3$)		70/70	0.051001	success
Random excursions variant ($x = 5$)		70/70	0.061150	success
Serial ($m = 16$)	P-value1	99/100	0.678686	success
	P-value2	99/100	0.719747	success
Linear complexity ($M = 1000$)		98/100	0.994250	success

The SP800-22 standard provides that at least 96 of 100 groups pass the test for each test item in order to achieve the SP800-22 standard requirements. During the test, some test items maybe only test the 69 or 70 among 100 groups and at least 65 and 66 among 69 and 70 groups respectively pass the test for each test item in order to achieve the SP800-22 standard requirements.

Tables 7 and 8 show that the CSS and the LCS can completely pass the SP800-22 standard test, however, the RC5 (RC5 test results not given) cannot completely pass, owing to the failure test item of overlapping template.

5.6. Analyses of speed, time, and storage space

In the WSNs, in addition to the above safety considerations, we must take into account some other issues including storage space, speed, time, and energy consumption.

The node energy consumption of the CSS is reflected in the run speed and the time of this encryption algorithm, as the energy consumption of nodes is reflected mainly in underlying communication, routing, sending, and receiving data. Table 9 shows only the test results of the CSS, RC5, and LCS on the emulator, as the other three encryption algorithms cannot run on the emulator for the node resource constraints. The test platform of the emulator is the CC2430 which is a system-on-chip (Soc) with an industry-standard enhanced 8051 MCU, 128 kB flash memory, 8 kB RAM, and running at 32 MHz.

From Table 9, we can clearly see that the CSS is far better than the RC5 and LCS in the space of variables or the running time and speed of the encryption algorithm.

Table 9. Test results for CSS, RC5 and LCS.

Algorithms / Test items	Time/(μs/byte)	Speed/(kB/s)	Space for global variables/byte	Space for local variables/byte
CSS	600	1.628	16	18∼20
RC5	876	1.115	168	88∼102
LCS	790	1.236	26	25∼41

6. Conclusions

In this paper we present a novel block encryption scheme based on the integer discretization of a chaotic map and the Feistel network structure and an S-box. The novel scheme is a fast, secure, low resource consumption algorithm and is suitable for WSN node encryption schemes. Experimental tests are carried out with detailed analyses, demonstrating that the new scheme has a large key space, very good diffusion and disrupt performance, strict avalanche effect, excellent statistical balance, and fast encryption speed, and the encryption scheme easily passes the SP800-22 test. Meanwhile, the analyses and the tests of speed, time, and storage space on the simulator platform show that this new encryption scheme is able to hide the data information of a node in WSNs well.

References

[1] Wang Y, Attebury G and Ramamamurthy B 2006 *IEEE Communications Surveys & Tutorials* **8** 2

[2] Sun L M, Li J Z, Chen Y and Zhu H S 2005 *Wireless Sensor Networks* (Beijing: Tsinghua University Press) p. 37 (in Chinese)

[3] Yang J Y 2007 *Research on Applications of Chaos Cryptography to Wireless Sensor Networks Security* Ph. D. dissertation (Chongqing: Chongqing University) (in Chinese)

[4] Wang S 2007 *The Ttheory and Application for Wwireless Sensor Networks* (Beijing: Beihang University Press) pp. 7–9 (in Chinese)

[5] Sun F Y and Lü Z W 2011 *Acta Phys. Sin.* **60** 040506 (in Chinese)

[6] Gu Q L and Gao T G 2009 *Chin Phys.* B **18** 84

[7] Li X C, Gu J H, Wang Y L and Zhao T H 2011 *Acta Phys. Sin.* **60** 030505 (in Chinese)

[8] Li J B, Zeng Y C, Chen S B and Chen J S 2011 *Acta Phys. Sin.* **60** 060508 (in Chinese)

[9] Zeng Z F, Qiu H M and Zhu L H 2007 *Application Research of Computers* **1** 140

[10] Liao X F, Xiao D and Chen Y 2009 *The Principle and Application of Chaotic Cryptography* (Beijing: Science Press) p. 1 (in Chinese)

[11] Xu S J, and Wang J Z 2008 *Acta Phys. Sin.* **57** 37 (in Chinese)

[12] Sun Y 2007 *Research and Implementation of Chaos Encryption* (Changsha: National University of Defense Technology) p. 10 (in Chinese)

[13] Hu J F and Guo J B. 2008 *Acta Phys. Sin.* **57** 1447 (in Chinese)

[14] Li W, Hao J H and Qi B 2008 *Acta Phys. Sin.* **57** 1398 (in Chinese)

[15] Chen S 2006 *Research on Chaos Encryption Theory and Key Technology for Wireless Micro-Sensor Network* Ph. D. dissertation (Chongqing: Chongqing University) (in Chinese)

[16] Tan Y J 2010 *Research on a Chaotic Block Cipher for WSNS* Ph. D. dissertation (Chengdu: University of Electronic Science and Technology of China) (in Chinese)

[17] Rivest R L 1995 *Dr. Dobb's Journal* **20** 146

[18] Rivest R L 1995 *Proceedings of Fast Software Encryption, Lecture Notes in Computer Science* (Berlin: Springer-Verlag) **1008** 86

[19] Yee W L, Jeroen D and Pieter H 2006 *ACM Transactions on Sensor Networks* **2** 65

[20] Rivest R L 1994 *Proceedings of the Second International Workshop on Fast Software Encryption* (New York: Springer-Verlag) p. 86

[21] Guido B, Luca B, Israel K Paolo M and Vincenzo P 2003 *14th IEEE International Conference on Application-Specific Systems, Architectures and Processors* (ASAP'03) (The Hague: The Netherlands) p. 423

[22] Rivest R L 1998 [EB/OL] http://people.csail.mit.edu /rives t/RC6.pdf

[23] Sun S L 2004 *Application Cryptography* (Beijing: Tsinghua University press) p. 23 (in Chinese)

[24] Buchholz J 2006 [EB/OL] http://buchholz. hs-bremen.de/aes/AES.pdf

[25] 2001 [EB/OL] http://csrc.nist.gov/publications/fips/ fips197/fips-197.pdf

[26] Zeis B and Edmister M 1998 [EB/OL] http://csrc.nist.gov/groups/ST/toolkit/documents /skipjack/skipjack.pdf

[27] Liu X C and Feng D G 2000 *Journal of Software* **11** 1299

[28] Peng J and Liao X F 2006 *Journal of Electronics and Information* **28** 4

[29] Mohammad P and Nevenko Z 2000 *Computers & Security* **19** 467

[30] Liu Y Z 2005 *Cryptography and Network Security — Principle and Practice* (Beijing: Publishing House of Electronics Industry) pp. 135–320 (in Chinese)

[31] Shannon C E 1949 *Bell System Technology Journal* **28** 656

[32] Preneel B 2001 [EB/OL] http://www.nist.gov/aes