A Fast Adaptive Stream Cipher Algorithm and Expanded Search

Xiaojing Hu
Automation Department, Xiamen University, Xiamen 361005, China
Email: kollzok@yahoo.com.cn

Lizhao Liu, Ying Wang and Maoqing Li
Automation Department, Xiamen University, Xiamen 361005, China
Email: {xiaojinghu1, yingwang03, maoqingli18}@yahoo.com.cn

Tianhua Zhang
Department of Electrical Engineering, Louisiana Tech University, Ruston 150001, USA
Email: tianhuazhang1@yahoo.com.cn

***Abstract*—To solve the problem of traditional stream ciphers can not transmit through the public channel and the construction of high-strength cipher key generator, the paper presents the minimum power clock function driven by a parallel clock-control sequence and chaotic cascade function made of multiple Logistic function, which are used to built the self-reference model and chaotic module; it designs an adapt controller independent of the key generator and encrypt module, achieve controlling encryption and transmission process by multi-threaded code; it makes a high-strength key generator with sure differential transformation and pseudo-random horizontal level disturbance wheel key method of AES and S-box of Camellia, which makes the sub-key and the key generator's internal station become vogue and uncertain within controlled area that does not depend on the secret channel, the model achieves an exponential growth in the key space and the same application scope of traditional stream cipher model. The new protocol model features can be seen under the actual operation of modifying RC4 and real-time RC5 algorithm.**

***Abstract*—Stream cipher; Minimum power clock function; Gaussian chaos cascade function group; Block ciphers; S-box; Pseudo-random horizontal level disturbance**

I. INTRODUCTION

Ciphers in cipher stream can be divided into Synchronous Stream Cipher, SSC and Self-Synchronous Stream Cipher, SSSC. SSC is widely used at present. The core of stream cipher is the key generator. In synchronous stream cipher, the generation of key stream is independent of the plaintext and ciphertext, see Figure 1, encryption transformation E and decryption transformation D are the time-varying function, the time variability is guaranteed by the memory file of encryption and decryption device. Figure 2 is the key stream generator model, in which g is a function of generating the key stream, f is a state function, g using the key seed k as input and produce an output key stream k1k2 … . As long as the transmitter and receiver have the same key and the internal state, the same key stream can be generated, then the key generator of both the transmitter and receiver is synchronized. Once out of sync, decrypt failed immediately [1] ~ [5]. SSSC ommited.

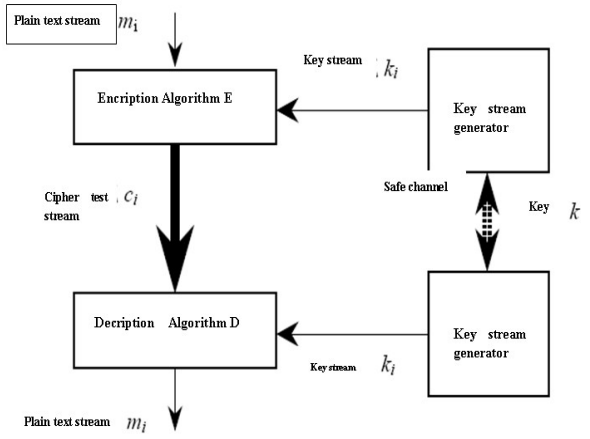


Figure 1. Synchronous Stream Cipher

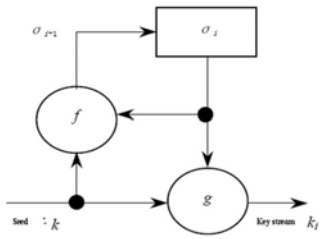


Figure 2. Key Stream Generator

The current construction method of key generator may be roughly divided into four categories: information theory approach, system theory approach, complexity theory approach and stochastic method [6] ~ [8]. In this paper, the author takes system theory and the principle of diffusion chaos of information theory as theoretical foundation, with adaptive control technology and chaotic mathematics to build SSC and SSSC protocol model and the corresponding key generator which can be easily realized and does not depend on secret channel. Since

doi:10.4304/jsw.6.10.1961-1968

they share the same principle, we only take SSC as an example to give a specific introduction. [9] ~ [13]

II. CONSTRUCTION OF MINIMUM POWER CLOCK FUNCTION AND GAUSSIAN CHAOS CASCADE FUNCTION GROUP

$$T=\{T_0,...T_k....Tn\}=\{t_0,...L\ (t_k,t_s)\ +W\bullet t_s\bullet t_k\bullet e^{-t_k\cdot w_k}\bullet COS(W_k\bullet t_s),...Tn\}$$

$$W_k=w_0\oplus X_{k-1}\oplus Y_{k-1}\oplus Z_{k-1}\oplus T_{k-1}$$

$$T_0=T(t,w_0)=\begin{cases}t_{last}, w_0=1\\ t_{now}, w_0=0\end{cases}$$

$$w_0=\begin{cases}0,(w_{1c}-E(w_1)\neq 0, w_{1c}-w'_{1c}\neq 0)\\ 1,(w_{1c}-E(w_1)=0, w_{1c}-w'_{1c}=0)\end{cases}$$

When $w_0=0$, t is updated to the current time t_{now}, when $w_0=1$, t maintains the latest value t_{last} after last update. w_t is the *Logistic* sequence of the first stage N, w_{1c} is the encryption result of w_t through encryption algorithm E, w'_{1c} is the return parameter of decryption end. The value of t_{last} and t_{now} are determined by clock sequence output function Ts. It is easier for clock output sequence to control the linear complexity than alignment sequence and non-linear combined sequences, by the construction of stimulus module through clock output signal controlling clock function, better pseudo-random and controlling can be achieved.

Chaotic sequence taking *Logistic* to generate cascade function:

$$E(f,K)=\beta\int_{\Omega}(f-g)^2dx+\int_{\Omega-K}|\nabla f|^2dx+\alpha|K|$$

$$f=Xn+1\oplus Yn+1\oplus Zn+1, g=Xn-1\otimes Yn-1\otimes Zn-1$$

$$\beta=t_0\otimes t_{k-1}, \alpha=t_{k-1}\oplus x_{k-2}, \Omega=\{x_1....x_{n-1}\}, \Omega-K=\{x_{n-k}...x_{n-1}\}$$

$$|K|=\sum_{k=1}^{k}X_k\bullet\sum_{k=1}^{k}Y_k\bullet\sum_{k=1}^{k}Z_k$$

$$\begin{cases}X_{n+1}=T_0\bullet X_n(1-X_n)\\ Y_{n+1}=X_n\bullet Y_n(1-Y_n)\\ Z_{n+1}=Y_n\bullet Z_n(1-Z_n)\end{cases}$$

The first-level key space is consist by [X(0), [T(t_0,w_0)], the second-level key space by [Y(0), X(0)], the third-level by [Z(0), Y(0)], in which, X(0), Y(0), Z(0), T(t_0, w_0) are the initial value of each map respectively. The genus-randomness, ergodicity and broadband nature makes it difficult for the analysis to find time-domain and frequency-domain characteristics of encrypted signals. Chaotic signal class, making analysis difficult are. By constructing a chaotic cascade function, the single chaotic function key space can be exponentially enhanced. [14]~[18]

III. CONSTRUCTION OF ADAPTIVE SSC MODEL AND KEY GENERATOR

A. Adaptive Control Flow Chart and Reference Model Design

Adaptive control technology can realize signal self-detection and self-adjusting. [19] ~ [21] The adaptive controller initialization within the control of continuous or intermittent output with automatic detection and adjustment function of the control signal, through the design of reference models or self-tuning controller module can be achieved on the output or received signal real-time adjustment and dynamic match. Adaptive encryption control principle is as Figure 3:

First initialize the clock module and the clock stimulus module as a self-reference model, since the self-reference model will reconstruct when the detective signal received from the self-detection module does not match, and the re-constructed reference model is not dependent on external stimulation, which depends only on the initial algorithm $\mathcal{E}$. This means that as long as both encryption and decryption have the same reference model, after the same initialization, they can always get synchronous control signal. For example, the use of the two CMOS unit can keep output synchronized at $k\bullet 10^8/S$.

Take the output signal from the self-reference model as the first stage parameter of chaos cascade module, the output signal of the first stage of chaotic module as the input signal of self-tuning module and at the same time, as the input signal of the second stage *Logistic* generator; the output signal of the second stage *Logistic* generator as the input signal of the third stage or input signal of key generator; the output signal of the third stage and key k together as the initial key of key generator.

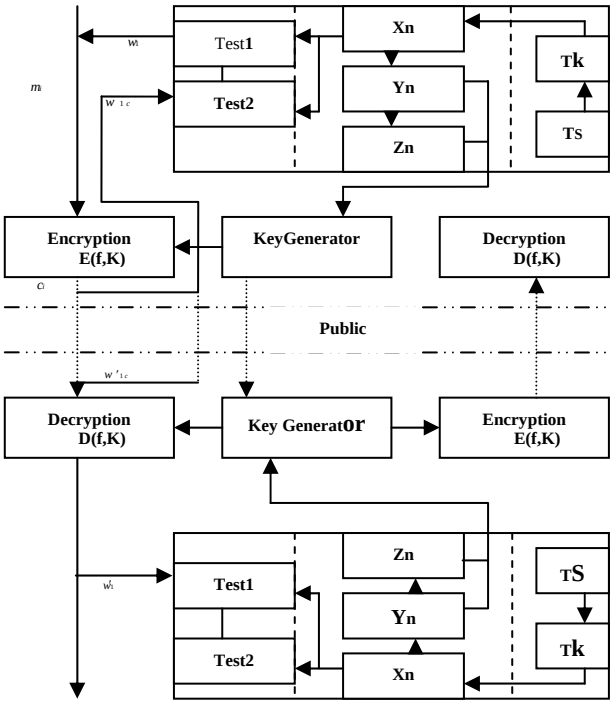


Figure 3. SSC Adaptive Stream Cipher Frame

The self-detection module consists of two detectors, which are responsible for adding state value w_t to the plaintext and testing encrypted state value w_{1c} and channel feedback state value w'_{1c}. When the encrypted value and the channel feedback state values are abnormal, feedback should be done to the reference model, then the self-reference model will update with a new reference model according to the current state value of initial

algorithm, thus to update all the output signals of chaotic module, key generators will also update new key stream without changing of key k.

The receiver uses the decoder to return w'_{1c} to the sender first to confirm the acceptance of the cipher, at the same time offers w'_{1c} evaluation to the sender for the channel safety index testing thus to determine whether there is need to resend the cipher or renew the reference model. Receiver, with the key k got from the sender through the public channel transmission and the synchronize chaotic sequence key stream K generated by its synchronization reference model, and then get the plaintext m' and test value w'_t through decoder D , if w'_t matches with its own test value w_t of the first stage Xn sequence of synchronization chaotic sequence with one's own first-level sequence, then it is confirmed that the received ciphertext is correct and decrypted successfully, if else, resend or renew reference model should be requested. The realization code and data format of self-adaptive SSC algorithm is shown in Figure 4 (each module operates in a separate thread to make sure the interrelated disruption of software is minimal)

```
public class Sende_Code
{ public static void main(String args[])
  Codemaker codemaker=new Codemaker();
  codemaker.selfModelThread.start();
  codemaker.codeInOutThread.start();}
  class Codemaker implements Runnable
   { int countzero, countone, ki, mi,;
        .............
    Codemaker()
     { selfModelThread=new Thread(this);
      codeInOutThread=new Thread(this); }
        .............
    public void run()
     { for(int countone=1,true; countone++)
      setDecision(countone);
      if (message==SUCCESS)
      return; }
}          .............
public synchronized void setDecision(int countone)
{ if (Thread.current Thread()=selfModelThread)
   {        ..................................
              while(Testone&&Testtwo)
            { Clock_MK(t);
              Logistic_MK(t); }
   }
  if (Thread.current Thread()=codeInOutThread)
   {        ..................................
              while(Testone&&Testtwo)
            { Clock_MR(t);
              Logistic_MR(t);}
   }        ..................................
 if (Thread.current Thread()=codeInOutThread)
{  for(int countzero=0;ki&&mi;countzero++)
   {keyFlow_maker();
    send_Message;}
}
}
```

```
%------------Process_object--------------
%T(1)=0;
%T(2)=5*b_3*y4(1)/b_2;
%sum_out=0;
%if k>1
%    for i=1:k
%        sum_out=sum_out+y4(i);
%    end
%    T(k+1)=(1*b_3*sum_out+0.5*b_1*T(k)-T(k-1))/b_2;
%end
%--------------study_Process_4&3--------------------
d4(1)=0;
d4(k)=(1.8-y4(k))
e(k+1)=d-y4(k);
ee(k+1)=e(k+1)-e(k);
for i=1:5
    for j=1:5
        d_o(i,j)=d4(k)*y3(i,j);
        d3(i,j)=d4(k)*o(i,j) ;
    end
end
%--------------study_process_2-------------------
d2=zeros(size(y2));
for j=1:5
    for i=1:5
        d2(1,j)=d2(1,j)+d3(j,i)*y2(2,i)*y2(1,j);
        d2(2,j)=d2(2,j)+d3(i,j)*y2(1,i)*y2(2,j) ;
    end
end
for i=1:2
    for j=1:5
        d_m(i,j)=d2(i,j)*2*(y1(i)-m(i,j))/q(i,j)^2;
        d_q(i,j)=d2(i,j)*2*(y1(i)-m(i,j))^2/q(i,j)^3
    end
end
%--------------modify parameters-----------------?
for i=1:2
    for j=1:5
        f_m(i,j)=sign(d_m(i,j)*d_m_1(i,j));
        f_q(i,j)=sign(d_q(i,j)*d_q_1(i,j));
        n_m(i,j)=2^f_m(i,j)*n_m(i,j);
        n_q(i,j)=2^f_q(i,j)*n_q(i,j);
        m(i,j)=m(i,j)+n_m(i,j)*d_m(i,j);
        q(i,j)=q(i,j)+n_q(i,j)*d_q(i,j);
        d_m_1(i,j)=d_m(i,j);
        d_q_1(i,j)=d_q(i,j);
    end
end
for i=1:5
    for j=1:5
        f_o(i,j)=sign(d_o(i,j)*d_o_1(i,j));
        n_o(i,j)=1.1^f_o(i,j)*n_o(i,j);
        o(i,j)=o(i,j)+n_o(i,j)*d_o(i,j);
        d_o_1(i,j)=d_o(i,j);
    end
end
k=k+1;
end ...
```

Figure 4. Self-adaptive SSC Code and Data Format

B. Key Generator Pseudo-random Horizontal Level Disturbance Design

There exist a fundamental relation between the order of differentiation of an operator, its code space and the required accuracy. For smaller s the Fourier transform of the kernel increases in width, at a certain space giving rise to aliasing. In theory this occurs at all space due to the infinite extent of the exponential function, but it becomes apparent at smaller space. The 'leaked' information is folded back, in theory even from all further periods as well, so the amplitude no longer represents the accurate value of 'pure' differentiation. We consider the power spectrum, i.e. the square of the signal. The aliasing differential operator integral operator and the error can be defined as the relative integrated energy of the aliased frequencies over the total energy that could do the

operation on the data as it for the mapping image of the database, so the following operation is done corresponding to the feature extracting and encryption to select the key-point information of the Stream Cipher [22]:

$$c(\vec{x},t) = c(\|\nabla L(\vec{x},t)\|) = e^{-\left(\frac{\|\nabla L\|^2}{k^2}\right)},$$

$$\frac{\partial L}{\partial s} = \nabla \cdot (c\nabla L),$$

$$E_{xy} = \sum_{u,v} W_{u,v}[xI_x + yI_y + O(x^2,y^2)]^2 \approx (x,y)M(x,y)^T$$

$$error(n,\sigma) = \frac{\Delta E_n(\sigma,\omega)}{E_n(\sigma,\omega)} = \frac{\int_\pi^\infty \omega^{2n} e^{-\sigma^2\omega^2} d\omega}{\int_0^\infty \omega^{2n} e^{-\sigma^2\omega^2} d\omega},$$

$$\sigma_{i,j} = \frac{\frac{1}{n}\sum_{l=-k}^{k}\sum_{m=-k}^{k}(p_{l,m}-\alpha)(q_{i+l,j+m}-\beta)}{\sqrt{\sigma(p)\sigma(q)_{i,j}}}.$$

$$\begin{aligned} C_1(x,y) &= \hat{I}_x^2\hat{I}_y^2 - (\widehat{I_xI_y})^2 \\ &= \sum_{x_1,y_1} g(x_1,y_1) I_x^2(x-x_1,y-y_1) \sum_{x_2,y_2} g(x_2,y_2) I_y^2(x-x_2,y-y_2) \\ &\quad - \left[\sum_{x_3,y_3} g(x_3,y_3) I_x(x-x_3,y-y_3) I_y(x-x_3,y-y_3)\right]^2 \\ &= \sum_{x_1,y_1}\sum_{x_2,y_2} g(x_1,y_1) g(x_2,y_2) \left[I_x^2(x-x_1,y-y_1) I_y^2(x-x_2,y-y_2)\right. \\ &\quad \left. - I_x(x-x_1,y-y_1) I_y(x-x_1,y-y_1) I_x(x-x_2,y-y_2) I_y(x-x_2,y-y_2)\right] \\ &= \frac{1}{2}\sum_{x_1,x_2,y_1,y_2} g(x_1,y_1) g(x_2,y_2) \left[I_x(x-x_1,y-y_1) I_y(x-x_2,y-y_2)\right. \\ &\quad \left. - I_x(x-x_2,y-y_2) I_y(x-x_1,y-y_1)\right]^2 \\ &= \frac{1}{2}\sum_{x_1,x_2,y_1,y_2} g(x_1,y_1) g(x_2,y_2) [(I_x(x-x_1,y-y_1), I_y(x-x_1,y-y_1))(-I_y(x-x_2,y-y_2) \\ &\quad I_x(x-x_2,y-y_2))]^2. \end{aligned}$$

The input terminus of key generator consists of the initial key K and the output signal of the second stage *Logistic* , the initial key and the third-level *Logistic* signal generate round keys, the round keys and the second stage chaotic signal generated key stream K through calculation . The structure is shown in Figure 5.

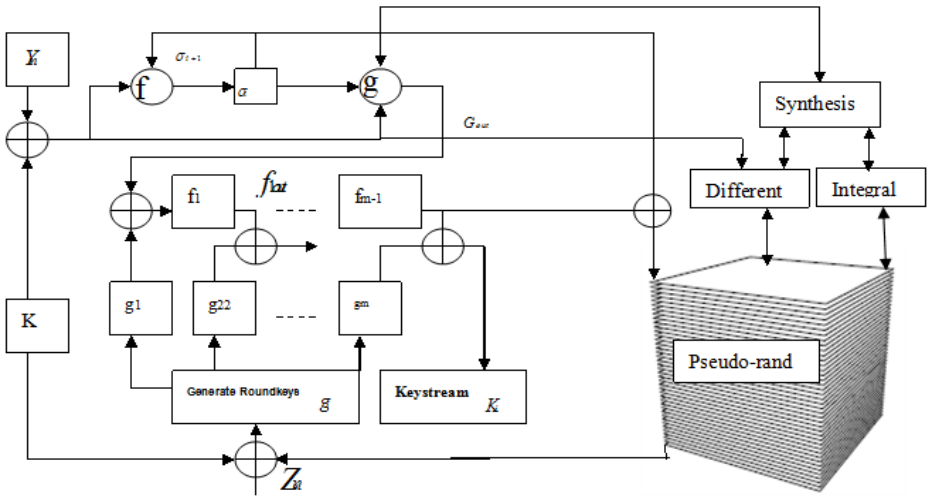
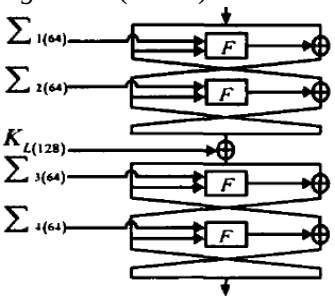


Figure 5. Pseudo-random horizontal level disturbance key generator

The output relationship of each module is as follows

$$\sigma_i = f[\sigma_{i+1}, (Y_n \oplus K)]$$
$$G_{out} = g[\sigma_i, (Y_n \oplus K)]$$
$$f_{1out} = f_1(g_{out} \oplus g_1)$$
$$f_{iout} = f_i(f_{i-1out} \oplus g_i),\ \text{m-2} \geq \text{i} \geq 2$$
$$K_i = f_{m-1} \oplus g_m$$
$$g_i = \text{mod}_i(K \oplus Z_n)$$



The calculation process is as follows [23] ~ [25]: (1) with initial data packet key and Yn bitwise XOR added transform the initial round keys. (2) Cycle m-2 round of data processing, 4 steps in a round

1) s box transform [26] ~ [28]: S box applying Camellia algorithm, it's a reversible transform on $GF(2^8)$, which can increase the safety and contribute to the hardware miniaturization of the algorithm. As it has been proved that the minimum value the maximum differential probability of the function on $GF(2^8)$ is 2^{-6}, and it is speculated that the minimum value of the maximum linear probability is also 2^{-6}. So choose the reversible function which can get the best differential and linear probability at $GF(2^8)$ as S box, meanwhile, the output bite of S box has higher-order Boolean polynomial which makes the high-order differential attack for the algorithm quite difficult. In addition, the complex expressions of S box at $GF(2^8)$ output and input functions makes the insert attack to the algorithm less effective. The bytes replacement of s box:

$$S: L \to L$$

$$(l'_{1(8)}, l'_{1(8)}, l'_{1(8)}, l'_{1(8)}, l'_{1(8)}, l'_{1(8)}, l'_{1(8)}, l'_{1(8)}) \to$$

$$(s_1(l'_{1(8)}), s_2(l'_{2(8)}), s_3(l'_{3(8)}), s_4(l'_{4(8)}), s_2(l'_{5(8)}), s_3(l'_{6(8)})$$

$$s_4(l'_{7(8)}), s_1(l'_{8(8)}))$$

$$s_1: y_{(8)} = s_1(x_{(8)}) = h(g(f(0xC5 \oplus x_{(8)}))) \oplus 0x6E$$

$$s_2: y_{(8)} = s_2(x_{(8)}) = s_1(x_{(8)}) <<< 1$$

$$s_3: y_{(8)} = s_3(x_{(8)}) = s_1(x_{(8)}) >>> 1$$

$$s_4: y_{(8)} = s_4(x_{(8)}) = s_1(x_{(8)} <<< 1)$$

$$f: L \to L$$

$$\begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \\ b_8 \end{pmatrix} = \begin{pmatrix} 0&1&0&0&0&1&0&0 \\ 1&0&0&0&0&0&1&0 \\ 0&0&1&0&1&0&0&1 \\ 0&0&1&0&0&0&0&1 \\ 0&0&0&1&0&0&1&0 \\ 0&1&0&0&1&0&0&0 \\ 1&0&0&0&0&0&0&1 \\ 0&0&0&1&0&1&0&0 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \end{pmatrix}$$

$$h: L \to L$$

$$\begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \\ b_8 \end{pmatrix} = \begin{pmatrix} 0&1&0&0&1&1&0&0 \\ 0&1&0&0&0&1&0&0 \\ 0&0&0&1&0&0&1&0 \\ 0&1&0&0&0&0&0&1 \\ 0&0&1&0&0&0&1&0 \\ 1&0&0&0&0&0&0&1 \\ 1&0&0&0&1&0&0&0 \\ 0&0&1&0&0&1&0&0 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \end{pmatrix}$$

$$g: L \to L$$

$$(b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8) = g(a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8)$$

$$(b_8 + b_7\alpha + b_6\alpha^2 + b_5\alpha^3) + (b_4 + b_3\alpha + b_2\alpha^2 + b_1\alpha^3)\beta$$

$$= 1 / ((a_8 + a_7\alpha + a_6\alpha^2 + a_5\alpha^3) + (a_4 + a_3\alpha + a_2\alpha^2 + a_1\alpha^3))$$

2) Line shift operation: take row cycle left shift operation;
3) Column confusion: take matrix multiply each column grouping;
4) Round key and operation: the initial key and Zn data block bitwise XOR, round keys are generated after expansion, with the current round keys and the current round of data packets by bit XOR

(3) The final round of transformation. It has 3 steps: 1) s box transforms; 2) the line shift operation; 3) round key addition operation. Algorithm is as mentioned above.

IV. PRACTICAL APPLICATION OF THE MODEL

RC4 algorithm source code was publised in 1994 in Cypherpunks's mailing list. RC4 can be simply described as: non-linear part is an8-in-and-8-out S-box, is the displacement from 0 to 255, and the replacement is a key function of a variable length. We directly use RC4 to encrypte a binary image, see Figure 6

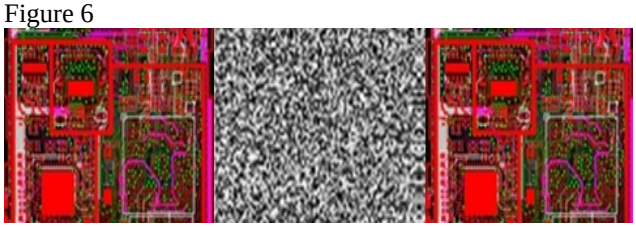


Figure 6. RC4 Encryption Results

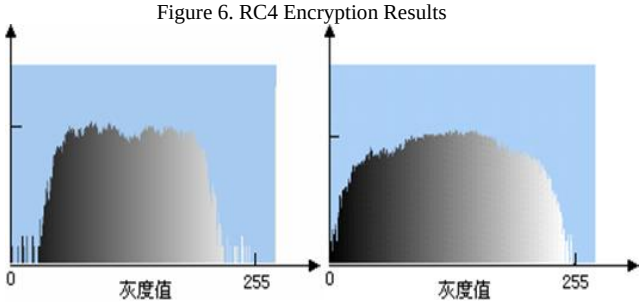


Figure 7. Encryption Histogram

Key generation algorithm and encryption algorithm are:

```
for (i= 0, i<255,i++)
{S[i] = i;j = 0;}
for (i= 0, i<255,i++)
{j = (j + S[i] + k[i ]) (mod 256);
swap (S[i], S[j]);}
int i = j = 0;
//for each message byte Mi//
i = (i + 1) (mod 256);
j = (j + S[i]) (mod 256);
swap(S[i], S[j]);
t = (S[i] + S[j]) (mod 256);
Ci = Mi XOR S[t];
```

Encryption results is shown in Figure 6 (middle); encryption effect histogram is in Figure 7 (left); when the decryption side access key K, through the decryption algorithm D, the image accessed is as in Figure 6(right).

Figure 8. Decryption Results

Revise RC4 algorithm according to self-adaptive SSC protocol model, the codes referred to Section 2 and 3 of this paper, the current PC clock is taken as the initial clock of reference model parameters in self-adaptive control part, encrypt the Figure 6 (left), the result is asFigure 8 (left); when decryption side has the synchronized self-adaptive module and obtains the key K, take out the decryption and get Figure 8 (middle); if the decryption side synchronized self-adaptation module does not work but still has a key K, the decrypted image is as Figure 8 (right) shows.

	Cycle count	Time(μS)	Pg(data)	Data/S	Stable-AA(ip)	Stable-BB(ip)
Key Generator	13659	1358.34	152	C9, T28	0. 6367	0. 41063
Encryption Progress	3012	274.31	70	163052	0. 8253	0. 69243
Decryption Progress	3032	280.22	73	163179	0. 8359	0. 77635
Total Progress	19703	1912.87	295	326231	0. 7737	0. 57282

Table 1 RC4 Operating Parameters

It can be seen that the improved RC4 algorithm based on SSC self-adaptive protocol model can not decrypt the original image with key k , decryption function E and key generator neither the bytes files. So we can say the key K can be transmitted in the public channel, a better randomness can be expected from the encryption histogram of the new model. The operating parameters of the process is shown in Table 1.

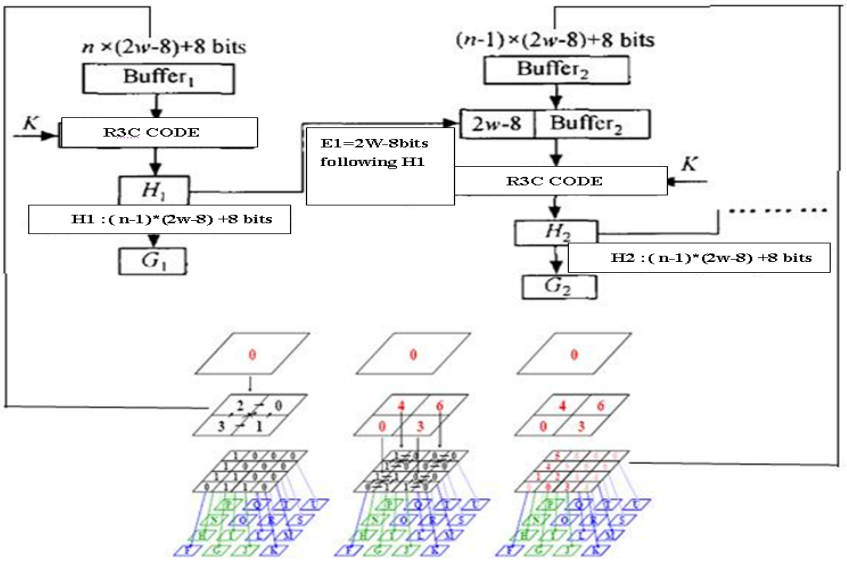


Figure 9. Improved RC5 Method

We also have done the improved RC5 algorithm that can be used in the real-time work, then we take an experiment on the new frame of the RC5 (Fig 9). When encrypt the index value of the leaf nodes, the ciphertext of previous node are taken as part of the plaintext of the next level, use RC5 keys to encrypt it together with the data of next leaf nodes. But the number of nodes and the encrypted results of the same level are put together into the encryption algorithm, that's to say, the encryption process of nodes number are taken as nested sub-algorithm to create multiple encrypted results. See Figure 5. Conduct a circulate encryption algorithm to m Buffer of this plaintext, use R3C encryption algorithm in each region. If the length of plaintext is between m-Buffer and m +1 Buffer, then RC5 encryption are done on the part beyond the m-Buffer separately. While here m may have different values, which makes it easier to control the size of the data. The operating results can be seen from the Fig10 and the parameters of the process is shown in Table 2.

Table 2 RC5 Operating Parameters

	Cycle count	Time(μ S)	Pg(data)	Data/S	Stable-AA(ip)	Stable-BB(ip)
Key Generator-A	8659	658. 34	722	C9, T28	0. 7467	0. 42636
Encryption Progress-A	1012	134. 31	34	193061	0. 9483	0. 76243
Decryption Progress-A	1072	156. 22	34	18159	0. 9389	0. 83635
Total Progress-A	10743	948. 87	790	136537	0. 8937	0. 67282
Key Generator-B	7328	623. 78	843	C5, T49	0. 7345	0. 4978
Encryption Progress-B	1235	198. 45	28	146575	0. 9934	0. 93684
Decryption Progress-B	1127	183. 56	29	17239	0. 9948	0. 98467
Total Progress-B	11038	964. 03	894	158344	0. 9473	0. 89426
Total Progress-	10782	956. 31	852	147294	0. 9059	0. 77375

```
void main()
{ WORD i, j, pt1[2], pt2[2], ct[2] = {0,0};
  unsigned char key[b];
  if (sizeof(WORD)!=4)
    printf("RC5 error: WORD has %d bytes.\n",sizeof(WORD));
  printf("RC5-32/12/16 examples:\n");
  for (i=1;i<6;i++)
    { /* Initialize pt1 and key pseudorandomly based on previous ct */
      pt1[0]=ct[0]; pt1[1]=ct[1];
      for (j=0;j<b;j++) key[j] = ct[0]%(255-j);
      /* Setup, encrypt, and decrypt */
      RC5_SETUP(key);
      RC5_ENCRYPT(pt1,ct);
      RC5_DECRYPT(ct,pt2);
      /* Print out results, checking for decryption failure */
      printf("\n%d. key = ",i);
      for (j=0; j<b; j++) printf("%.2X ",key[j]);
      printf("\n   plaintext %.81X %.81X  --->  ciphertext %.81X %.81X  \n"
             pt1[0], pt1[1], ct[0], ct[1]);
      if (pt1[0]!=pt2[0] || pt1[1]!=pt2[1])
        printf("Decryption Error!");
    }
}
```

It can be seen that the improved RC5 algorithm based on SSC self-adaptive protocol model also can not decrypt the original image with key k , decryption function E and key generator neither the bytes files. The key K can be transmitted from the public channel.

V. CONCLUSION

The reference model of self-adaptive controller takes the Minimum power clock function control module by a parallel clock-control sequence and chaotic cascade function made of multiple Logistic function, which is assigned a pair of encryption and decryption, through the same initialization of as the stimulus unit, the unit is theoretically synchronous, but no stability tests carried out on the actual system for a long period, and there should be a certain redundancy in the actual transmission between the sender and the receiver; key generator consists of the most safe and efficient parts of sophisticated block algorithm, AES and Camellia, but compares with the traditional stream cipher model, these parts have no advantage in speed. ith the public algorithm analysis being more sophisticated, the safety of the key generator will face new challenges. In this paper, the test of the modified version of RC4 and RC5 algorithm has been carried out, but some other algorithms, such as A5, E0, etc. also need to be modified to test the applicability

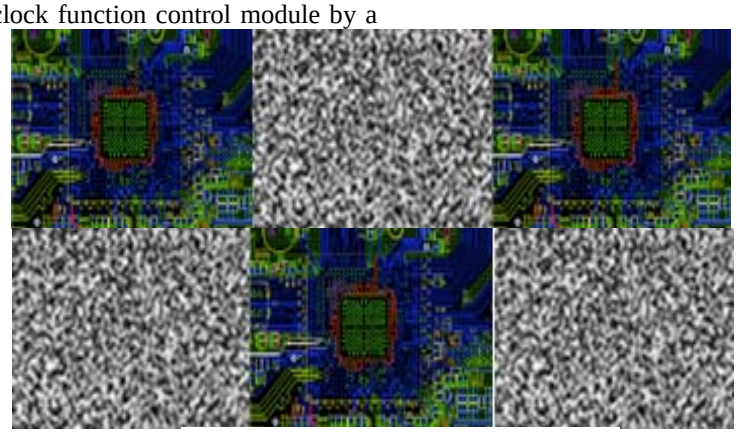


Figure 10. RC5 Encryption and Decryption Results

ACKNOWLEDGMENT

This work is supported by the Natural Science Foundation of Fujian Province（2010J01353） and Open Fund (201001) Funding from key lab of Spatial Data Mining and Information Sharing, Ministry of Education, Fuzhou University;Open Fund (BLISSOS2010102) Funding from key lab of of Brain Imitation Intelligent Systems in Xiamen University Fujian Province.

REFERENCES

[1] CE Shannon. Communication theory of secrecy systems[J].Bell System Technical Journal,1949,28(4):656-715.

[2] Deng-Guo Feng. Abroad Cryptography Research and Development [J]. Communications,2002,23(5):18-26.

[3] Linde Jing, LIN Bo-gang. Three cryptography: symmetric encryption, public key cryptography and quantum cryptography theory and techniques [J]. Telecommunications Technology,2003,43(3):6-12.

[4] Small deposit of Health, Xiao Guozhen. Stream Cipher and Its Application [M]. Beijing: National Defence Industry Press, 1994.

[5] Luoqi Bin, Zhang Jian. Current status and development of code [J]. Telecommunication Technology, 2003, and 43 (three) :6-12.

[6] Hu Yu-pu, Zhang Yuqing, Xiao Guozhen. Symmetric Cryptography [M]. Beijing: Mechanical Industry Press, 2002.

[7] C E Shannon. A Mathematical Theory of Communication[J]. Bell System Technical Journal, 1948,27(4):379-423,623-656.

[8] Luo Lan. Block cipher design and evaluation of Applications. Dissertation, University of Electronic Science and Technology .2009.

[9] International Organization for Standardization. ISO 7498-2: 1989, Information processing systems. Open Systems Interconnection. Basic Reference Model. Part 2: Security Architecture,1989

[10] International Telecommunication Union. CCITT Recommendation X.800 (1991), Data Communication Networks: Open Systems Interconnection (OSI); Security,

Structure and Applications Security Architecture for Open Systems Interconnection for CCITT Applications,1991
[11] International Organization for Standardization. ISO/IEC 10181.1: 1996, Information technology . Open Systems Interconnection. Security frameworks for open systems: Overview,1996
[12] International Organization for Standardization.ISO/IEC 10116:1997,Information Technology. Security Techniques. Modes of Operation for an n-bit Block Cipher,2nd edition,1997
[13] Zhou Hong, Xie-Ting Ling. Chaotic secure communication principles and confidentiality of [J]. Circuit and Systems Science, 1996, 1 (3): 59-60
[14] Cui Guangliang. Based on parameters and Perturbation theory of chaotic secure system design [J]. Shanghai Jiaotong University, 2004,38(11): 1818-1821.
[15] Aquatic, Yan-Feng Chen, Wu Min, and so on. A new principle of chaotic encryption system solutions [J]. Circuits and Systems, 2006, 11 (1): 100-102.
[16] Fan Yi, Xiong-Ying Liu, Qiu Shui-sheng. Chaotic encryption and conventional encryption secure communication system image composite [J]. Computer Engineering, in 2005, 31 (20): 44-45.
[17] Xu Yan. Based on Logistic Mapping BMP image encryption algorithm [J]. Modern computer, 2009,2 (4)
[18] A Community Yin Guofu. Chaotic system based on composite image encryption method [J]. Computer Applications, 2006,26 (4)
[19] Slottine J E, Li W. Applied Nonlinear Control [M]. Englewood Cliffs, NJ: Prentice Hall, 1991
[20] Zhang T, Ge S S, Hang C C. Design and performance analysis of a direct adaptive control for nonlinear systems [J]. Automatica, 35(10): 1809-1817, 1999
[21] Gavel D T, Siljak D D. Decentralized adaptive control: structural conditions for stability [J]. IEEE TransAutomat. Contr., 34(4): 413-426, 1989
[22] B. M. ter Haar Romeny, W. J. Niessen, J. Wilting, and L. M. J. Florack. Differential structure of images: Accuracy of representation. In Proc. First IEEE Internat. Conf. on Image Processing, pages 21–25, Austin, TX, November, 13-16 1994. IEEE.
[23] Huijuan, QIU Shui-sheng, Luminescence. AES algorithm based on chaotic map and image encryption scheme [J]. Computer Engineering, 2007,24 (4)
[24] National Institute of Standards and Technology (NIST). Federal Information Processing Standards Publication 197 (FIPS PUB 197): Specification for the Advanced Encryption Standard (AES), November 2001
[25] W.E.Burr, Selecting the Advanced Encryption Standard. IEEE Security and Privacy,March/April 2003 1(2):43-52
[26] Jing-Wei Liu, Bao-Dian Wei, Sun Rong, Xin-mei. A new and practical security encryption standard algorithm-Camellia algorithm. Electronics applications, 2003,11.
[27] Kazunlaro Aoki, Tetsuya Ichikawa, etc. Camellia:A 128-Bit Block Cipher Suitable for MultiplePlatforms.http://www.cosic.esat.kuleuven.ac.be/nessie/workshop/submissions/Cam elia. zip. 2001.9.26.
[28] Kazulnaro Aoki, Tetsuya Ichikawa, etc. Speciilcation of a 128-bit Block Cipher. https://www. cosic. csat. kuleuven. ac. be/nessie/workshop/submissions/Camellia. zip. 2000. 3. 10

Xiaojing Hu 12-5-1962, xiamen city, fujian province, China. PHD candidate of Xiamen university, majored in automation ,system engineering, information science and technology college. Research field: electric information system analysis and design, system engineering. He made the paper such as A New Adaptive SSC and SSSC Stream Cipher Model Design and Implementation 2010 SMIS and A New Close-form Solution for Initial Registration of ICP 2010 AMR. He also has done the work of A Real-time 3D Collision Detection Encryption Algorithm Based On Improved RC5 and had the paper on the ASL journal.

Lizhao Liu 30-3-1983, xiamen city, fujian province, China. PHD candidate of Xiamen university, majored in automation ,system engineering, information science and technology college. Research field: chaotic modeling and control of unmanned airplane vehicle and information system, feature attraction and detection, scale space and multiscale technology.

He has done the China national 985 engineering process of unmanned airplane vehicle for the UAV\UAIS chaotic phenomenon analysis , UAV\UAIS chaotic modeling and control. He made the paper such as The Chaotic Characters and New Control Strategy of Unmanned Airplane Information System 2008 ISCID, The Chaotic Disturbance of UAV System's Communication And Coping Strategy 2008 ICCCAS. He also has done the work of grid behavior trust model and has the paper such as The Quantitative Assignment of The Grid Behavior Trust Model Based on Trusted Computing 2010 Wuhan university journal.Now he is doing the work of scale space and multiscale technology for the image analysis especially for the feature describtion definition detection and matching.

Tianhua Zhang 10-7-1983, ruston city, louisiana state, USA. PHD of Louisiana university, majored in electrical engineering, Research field: electric information engineering.

Ying Wang 13-5-1978, xiamen city, fujian province, China. PHD and professor of Xiamen university, majored in automation ,system engineering, information science and technology college. Research field: system engineering.

Maoqing Li 23-6-1954, xiamen city, fujian province, China. PHD supervisor and professor of Xiamen university, majored in automation ,system engineering, information science and technology college. Research field: system engineering.