

DESCRIPTION

SEQUENCE ENCRYPTION METHOD ACCOMPANYING ADJUSTABLE RANDOM RECONFIGURATION OF KEY

TECHNICAL FIELD

The invention relates to the field of sequence cryptography in information security cryptography, and mainly relates to a sequence encryption method accompanied with adjustable randomly-reconstructed key.

BACKGROUND OF THE PRESENT INVENTION

Encryption with the sequence cipher is a process of obtaining a cipher text by bitwise XOR of a pseudo-random sequence generated according to a seed key and a plaintext bit stream using a sequence cipher generator. The sequence cipher encryption method may encrypt plaintext of any length as long as the period of the generated sequence cipher is large enough. The sequence cipher encryption method is faster than a block cipher encryption method. The density of the sequence cipher encryption depends on the randomness of the sequence key generated by the sequence cipher generator. When the probability of acquiring the sequence key is small enough, the probability that the cipher is deciphered is small enough.

SUMMARY OF THE PRESENT INVENTION

The purpose of the present invention is to overcome the deficiencies in the prior art, and to provide a sequence encryption method accompanied with adjustable randomly-reconstructed key.

It is an object of the present invention to be accomplished by the following technical scheme. This sequence encryption method accompanied with adjustable randomly-reconstructed key implements sequence encryption/decryption accompanied with reconstructing a random sequence key based on a random binary sequence during encryption, the method comprising steps of:

DESCRIPTION

(1) dynamically constructing a seed random sequence B_i at beginning of each encryption through an existing random sequence, wherein the seed random sequence B_i is derived from a pseudo-random reconstruction of an alternative random sequence B^j , or is directly taken from the alternative random sequence B^j ; wherein B^j is a part of a contract of encryption and decryption parties, and the seed random sequence B_i is pseudo-randomly reconstructed or truncated from B^j starting from an agreed position according to an agreement;

(2) pseudo-randomly constructing a random key $\underline{B}$ using the seed random sequence B_i with a bit segment of a fixed or unfixed length; wherein the generation of the random key $\underline{B}$ is accompanied during encryption/decryption process and the random key $\underline{B}$ is generated step by step through pseudo-randomly reconstructing each bit in the seed random sequence B_i ; selecting a key bit segment for XOR in each step according the secret agreement of the encryption and decryption parties, wherein the key bit segment includes frequencies of XOR on a plaintext, a starting position of bit segment at each frequency and length of the bit segment, and a process of the pseudo-random reconstruction is determined by the secret agreement; and

(3) encrypting/decrypting a segment of the plaintext by the bit segment of the fixed or unfixed length, wherein each segment of the plaintext XORs with several corresponding key bit segments obtained by the pseudo-random reconstruction with a fixed frequency or variable frequency, an increase or no increase between frequencies and swapping or not swapping a cipher text between frequencies, to generate the cipher text.

The present invention has the beneficial effects of pseudo-randomly manipulating an encryption logic process by randomness of a random sequence to implement sufficient chaos and concealment of key construction, thereby blocking an approach from parsing a cipher text with a secret agreement privately agreed by the encryption and decryption parties.

BRIEF DESCRIPTION OF THE DRAWINGS

DESCRIPTION

FIG. 1 is a schematic diagram of pseudo-randomly constructing a seed random sequence by forward shifting of two alternate random sequences in accordance with the present invention;

FIG. 2 is a schematic diagram of a staggered random reconstruction state of a key $\underline{b}_s$ in accordance with the present invention;

FIG. 3 is a schematic diagram of an encryption process with a pseudo-random reconstruction instance in accordance with the present invention.

Reference signs: the key $\underline{b}_s$ in FIG. 3 is taken by truncating b_j to b_{j+L} bits from B by pseudo-random reconstruction; m_s is taken through forward shifting by L_s length; c_s is generated by XOR for the m_s in forward shifting by L_s length; E is an engine randomly generated for each encryption; and S is a feature number.

DETAILED DESCRIPTION OF THE PRESENT INVENTION

The present invention will be described in detail below with reference to the accompanying drawings. It can be understood by those skilled in the art that alternations and alterations of the solutions and concepts of the present invention will fall within the scope of the claims of the present invention.

A concept called pseudo-random reconstruction in the present invention is first introduced: a function F has a value domain which covers a random sequence or other determined range and arguments which can comprise a number of random values. A process for obtaining a new random sequence comprises obtaining the lengths (in bits) of bit segments, the binary start positions of the bit segments and other manipulation information of the random sequence by substituting a random value associated or unassociated with the random sequence into the function, and in turn performing reconstruction segment by segment with the bit segments determined by such start positions and lengths and other obtained bit segments.

DESCRIPTION

The purpose of the present invention is to implement sequence encryption by generating a seed random sequence through pseudo-random construction using a random number called as engine for an existing random number sequence, and in turn, obtaining random key bit segments from the seed random sequence through pseudo-random reconstruction step by step. The pseudo-random reconstruction process is accompanied throughout the encryption process, where in each step, the key bit segment selected by the random number obtained through the pseudo-random reconstruction in the previous step is used for XOR encryption with the plaintext bit segment in the current step. The key bit segments in the steps are combined into an encryption key at current time. Random reconstruction in each step may have a current random bit segment for which a feature number participates in the manipulation of the seed sequence, and the entire reconstruction process presents a chaotic morphology determined by the randomness of the seed sequence. A function sequence is formed by the random number sequence reconstruction function as the encryption process proceeds. The function sequence may be adjusted by the secret rules agreed by the encryption and decryption parties. An instant key sequence obtained by manipulating the random seed sequence using the function sequence presents randomness and concealment, and has the effect of approximate one chaotic codebook per time.

Another important purpose of the present invention is to take advantage of the transitivity of a XOR operation, such that in each XOR step, a multi-frequency XOR is performed by a random number of times determined by the value of the key bit segment in the previous step and the constructed corresponding plurality of key bit segments. This random multi-frequency XOR manipulated by the pseudo-random reconstruction enables the entire encryption process to present random nonlinear morphology with variable frequencies.

The two basic principles described above are further expressed as follows:

DESCRIPTION

Infinite pseudo-random reconstruction is carried out on the finite random sequence to construct an infinite random sequence. Let B be as follows: $b_1, b_2, b_3, \dots, b_n$ which is an infinite binary random sequence. Assuming $b_{n+1} = b_{i1}, b_{n+2} = b_{i2}, \dots, b_{2n+1} = b_1, b_{2n+2} = b_2, \dots$, B is reconstructed with a random number sequence $L = L_1, L_2, \dots$ to obtain $\underline{B} = \{b_{i1+1}, b_{i1+2}, b_{i1+3}, \dots, b_{i1+L1}\}, \{b_{i2+1}, b_{i2+2}, b_{i2+3}, \dots, b_{i2+L2}\}, \dots$, where $L_1, L_2, \dots$ are constructed based on the value of the bit segment of $\underline{B}$ in the previous step. Thus, $\underline{B}$ is an infinite random sequence $\underline{b}_1, \underline{b}_2, \underline{b}_3, \dots$. This process, called as pseudo-random reconstruction in the present invention, is accompanied in the XOR process of $\underline{B}$ and plaintext to implement chaos of a run of elements in the random sequence $\underline{B}$. The run of elements in the random sequence $\underline{B}$ is more chaotic and has concealment when above $\underline{B} = \{b_{i1,1}, b_{i1,2}, b_{i1,3}, \dots, b_{i1,L1}\}, \{b_{i2,1}, b_{i2,2}, b_{i2,3}, \dots, b_{i2,L2}\}, \dots$ are selected by pseudo-random hopping or no hopping with alternative bit segments $B^j = \{b_{i1,1}^j, b_{i1,2}^j, b_{i1,3}^j, \dots, b_{i1,L1}^j\}, \{b_{i2,1}^j, b_{i2,2}^j, b_{i2,3}^j, \dots, b_{i2,L2}^j\}, \dots$ of one or more random sequences and then are each paired orderly to XOR.

When a certain bit segment of plaintext $m_s = \{m_{s1}, m_{s2}, m_{s3}, \dots, m_{sL}\}$ is multi-XOR with infinite $\underline{b}_{s1} = \{b_{i1}, b_{i2}, b_{i3}, \dots, b_{iL}\}, \underline{b}_{s2} = \{b_{j1}, b_{j2}, b_{j3}, \dots, b_{jL}\}, \dots$, $c_s = (m_s \oplus \underline{b}_{s1}) \oplus \underline{b}_{s2}, \dots$, the plaintext can be recovered by the multi-XOR of above result and the keys $(\underline{b}_{s1}, \underline{b}_{s2}, \dots)$, $m_s = (c_s \oplus \underline{b}_{s1}) \oplus \underline{b}_{s2}, \dots$. When the frequency of the multi-XOR is determined pseudo-randomly by the random value of key bit segment or associated bit segment in the previous step and is randomly distributed in the encryption process, the XOR process for the plaintext presents a nonlinear morphology with variable frequencies.

In order to ensure the concealment and chaos degree of the keys, the present invention provides a method of pseudo-random reconstruction of a seed random sequence B in which a drive random number E privately agreed by the encryption and decryption parties participates. With the method, it makes the encryption key more chaotic and hidden, while the seed random sequences employed in encryption at different times on basis of the same alternate random sequence present randomness

DESCRIPTION

variation. The present invention also provides the feature number S specific to each encrypted user to participate in above pseudo-random reconstruction. This makes the seed random sequences of different users on basis of the same alternate random sequence present randomness variation.

The pseudo-random reconstruction method of the present invention is applicable to both the pseudo-random reconstruction process of the seed random sequence B and the pseudo-random reconstruction process of the key $\underline{B}$, and the pseudo-random reconstruction can employ any of the following methods (1) through (13) according to the density needs:

(1) defining $B^j = \{b^j_1, b^j_2, \dots, b^j_n\}$ as an alternative random sequence and $B_i = \{b_{i,1}, b_{i,2}, \dots, b_{i,n}\}$ as a seed random sequence (i may be omitted when there is only one seed random sequence), $\underline{B} = \{\underline{b}_1, \underline{b}_2, \dots, \underline{b}_n\}$;

(2) determining a starting bit b^0_0 (or b_0) in an agreed manner (e.g., by an operation participated by the E and/or S) and truncating the target random sequence B (or $\underline{B}$) by forward/backward shifting any source random sequence B^0 (or B) by a fixed length;

(3) obtaining the target random sequence B (or $\underline{B}$) by orderly paring segments from more than one source random sequences B^j (or B_i) to XOR, where the segments are obtained by forward/backward shifting the source random sequences by the fixed length with the start bit b^j_0 (or $b_{i,0}$) determined in the agreed manner (e.g., by an operation participated by the E and/or S) respectively;

(4) obtaining the target random sequence B (or $\underline{B}$) by paring segments from one or more source random sequences B^j (or B_i) one by one to multi-frequency XOR, where the segments are obtained by forward/backward shifting the source random sequences by the fixed length with the start bit b^j_0 (or $b_{i,0}$) determined in the agreed

DESCRIPTION

manner (e.g., by an operation participated by the E and/or S) respectively;

(5) after determining the value $b^{0'}_0$ (or b'_0) of the start bit of the segments in the agreed manner, for any source random sequence B^0 (or B), performing the following operations according to the value $b^{0'}_{s-1}$ (or b'_{s-1}) of the target random bit segment or associated segment (e.g., a corresponding bit segment of the cipher text) in the previous step: ① determining the start bit p of the target bit segment b^0_s (or b_s) in the source random sequence in the current step with a positioning function P ; ② determining the length l of the source random bit segment b^0_s (or b_s) and the target random bit segment b^0_s (or b_s) in the current step with a length determining function L ; and ③ substituting the position and/or length in above (1) with the start position and/or length of the random source bit segment pseudo-random constructed in ① and/or ② to truncate the target random bit segment;

(6) after determining the value $b^{j'}_0$ (or b'_j) of the start bit of the segments in the agreed manner, for more than one source random sequences B^j (or B_i), performing the following operations according to the value $b^{j'}_{s-1}$ (or $b'_{i\ s-1}$) of the target random bit segment or associated segment in the previous step: ① determining the start bit b^j_s (or $b_{i,s}$) in each of the source random sequences with the positioning function P respectively; ② determining l of each of the source random bit segments in the current step with the length determining function L ; and ③ substituting the position and/or length in above (2) with the start position and/or length of each of the random source bit segments pseudo-random determined in ① and/or ② to orderly pair each of segments one by one to perform XOR thereby obtaining the target random bit segment b_s (or $\underline{b_s}$);

(7) after determining the value $b^{j'}_0$ (or b'_j) of the start bit of the segments in the agreed manner, for more than one source random sequences B^j (or B_i), performing the following operations according to the value $b^{j'}_{s-1}$ (or $b'_{i\ s-1}$) of the target random bit

DESCRIPTION

segment or associated segment in the previous step: ① determining the start bit b_s^j (or $b_{i,s}$) in each of the source random sequences with the positioning function P respectively; ② determining l of each of the source random bit segments in the current step with the length determining function L ; ③ determining frequency value f^j (or f_i) with a frequency determining function F ; obtaining the target random bit segment by performing the processes in ④ and ⑤ below: ④ selecting the source random bit segment $b_s^{j,1}, b_s^{j,1+1}, b_s^{j,1+2}, \dots, b_s^{j,1+f_j}$ (or $b_{i,s}^1, b_{i,s}^{1+1}, b_{i,s}^{1+2}, \dots, b_{i,s}^{1+f_i}$) by the manner of forward shifting each of the source random sequences or the manners in above ① and/or ② substituting those in (1) or (2) to orderly pair each of the segments one by one to perform the multi-frequency XOR, and the result being taken as the target random bit segment b_s (or $\underline{b}_s$); or ⑤ for each of the source sequences, selecting the frequency f^j (or f_i) for XOR by ③, and then selecting the source random bit segment $b_s^{j,1}, b_s^{j,1+1}, b_s^{j,1+2}, \dots, b_s^{j,1+f_j}$ (or $b_{i,s}^1, b_{i,s}^{1+1}, b_{i,s}^{1+2}, \dots, b_{i,s}^{1+f_i}$) by the manner of forward shifting or the manners in above ① and/or ② substituting those in (1) or (2) or (3) to pair each of the segments one by one to perform the multi-frequency XOR (the XOR frequencies for each of b_s^j (or $b_{i,s}$) are different), and the result being taken as the target random bit segment;

(8) a further pseudo-random reconstruction step of enhancing chaos degrees for pseudo-random reconstruction in (4), (5) and (6): constructing a bit hopping function J (whose argument is l random values determined by the key bit segment or associated bit segment in the previous step, with the value range being l $J(b_{i,s-1}^j)$ or $J(b_{i,s-1}^j)$, $J(b_{i,s-1}^j)$ or $J(b_{i,s-1}^j)$, $0 \leq J(b_{i,s-1}^j)$ or $0 \leq J(b_{i,s-1}^j)$) for selecting the bit segment of the source random sequence in the current step, starting from the source random sequence position p determined by P , according to the hopping function, the bits of the key bit segment in the current step being extracted bit by bit with hopping (when $J(b_{i,s-1}^j) > 0$) or without hopping (when $J(b_{i,s-1}^j) = 0$) until the length l of the target bit segment in the current step determined by L is reached;

(9) another enhanced pseudo-random reconstruction to determine the current step

DESCRIPTION

source random bit segment b_{s-1}^j or b_{is-1} or $b_{s-1}^{j,fj}$ according to one of the manners in above (4) or (5) or (6) or (7) to, frequency by frequency, select the source random sequence bit segment to perform XOR on the target bit segment one by one;

(10) a further enhanced pseudo-random reconstruction operation for the pseudo-random reconstruction in above (8): in encryption, the target bit segment after XOR per frequency (in decryption, the source bit segment before XOR per frequency) being swapping by the pairing bit determined by the pairing function C according to the random value of the key bit segment or associated bit segment in the previous step;

(11) a further enhanced pseudo-random reconstruction form for the pseudo-random reconstruction in above (9): for each frequency, determining the increase value of the current frequency key bit segment by the extension function A according to the value of the frequency key bit segment or associated bit segment in the previous step, where the source bit segment increases correspondingly with the increase value, and so on to increase the length of bit segment for XOR frequency by frequency; after the operation of the current step is completed, determining the length of the target bit segment in the current step by the initial bit segment length as well (the starting position of the target bit segment in the next step is still stepped by the target bit segment of the first frequency) and backfilling the increase operation part of the target bit segment to the corresponding position of the source sequence; as a result, the increased bits have been subjected to several XOR pretreatments before the next XOR, since the pseudo-random reconstruction of these XOR pretreatments in the decryption process is still determined by the value of the same key bit segment or corresponding bit segment, its inverse operation process is an isomorphic reverse process; this enhanced form enabling the accompanied pseudo-random reconstruction to be closer to the effect of one chaotic codebook per time;

(12) a different form of flexible pseudo-random reconstruction with same

DESCRIPTION

principle for the pseudo-random reconstruction in (4), (5), (6), (7), (8), (9), (10): the bit segment determined by P, L, F, J, C and A are changed to be extracted from another random sequence or extracted from a different source random sequence; the effect of constructing the random key in such a random bit still has concealment and chaotic characteristics;

(13) in above (4), (5), (6), (7), (8), (9), (10), the P, L, F, J, C and A functions may add S as an argument to implement a pseudo-random reconstruction with S features, where S may be one or more constants, variables of a function, or a combination thereof.

(14) the method for obtaining each bit segment of the target sequence can be implemented by combining above pseudo-random reconstruction manners in an agreed manner.

The encryption process of the present invention is carried out as follows:

(S0.1) pre-generating one or more groups of alternative seed random sequences:

$B^1: b^1_1, b^1_2, b^1_3, \dots, b^1_n; B^2: b^2_1, b^2_2, b^2_3, \dots, b^2_n;$

(S0.2) the agreed physical feature quantity S (can be one element, multiple elements, or the integration of multiple elements in a constant set and/or variable set) being used as the feature weight of the encrypted object to participate in the operations of P, L, F, J, C, and A, where a number of drive engine random numbers E1, E2, ... that match the alternate random sequences are generated (randomly selected from generated sets) for each encryption, as a trigger element for the pseudo-random construction (determining the starting state of the pseudo-random construction);

(S0.3) pseudo-randomly-reconstructed seed random sequence $B = \{b_1, b_2, b_3, \dots, b_s, \dots, b_n\}$ (which may be one or more sequences and can be

DESCRIPTION

pseudo-randomly constructed one by one with any of the following methods when more than one seed random sequence is used):

after locating s_0 with E, constructing $b_i = b_{(s_0+i) \% n}^J = 1, 2, 3, \dots, n$ by forward shifting (1)

or

after locating s_0 with E, randomly constructing $b_s = \text{Build}(b_{(s-1)}^J) s = 1, 2, 3, \dots, n$ (1')

or

after locating s_0 with E, randomly multi-frequency constructing $b_s = \text{Build}(b_{(s-1)}^{fJ}) s = 1, 2, 3, \dots, n$ (1'')

where:

$b_{(s_0+i) \% n}^J$ represents one of the pseudo-random constructions in (1), (2), (3), and $\%n$ represents continuing from the start after forward shifting to the end of the random sequence;

$\text{Build}(b_{(s-1)}^J)$ represents pseudo-randomly constructing the seed random bit segment by arguments of P, L, F, and J obtained by one of (4), (5), (6), (7), (8), (9), (10), and (11) in conjunction with S, E^J ;

$b_s = \text{Build}(b_{(s-1)}^{fJ})$ represents pseudo-randomly multi-frequency constructing the seed random bit segment by arguments of P, L, F, and J obtained by one of (4), (5), (6), (7), (8), (9), (10), and (11) in conjunction with S, E^J ;

(S1) calculating L_s :

(S1.1) when the plaintext stepped to L_s is less than or equal to the remaining plaintext length,

$$L_s = L(S, b'_{s-1}), \quad (2)$$

where L is a function of calculating a new bit segment length, for $s=1$, taking any b'_0 with $L(S, b'_{s-1})$ being not 0 (e.g., $b'_{(E+S) \% n}$);

b'_s represents the random value carried by the bit segment b_s (the same below);

DESCRIPTION

(S1.2) when the plaintext stepped to L_s is greater than the remaining plaintext length, the processing for the last bit segment is performed:

$$L_s = \text{remaining plaintext length} \quad (3)$$

(S2) calculating

$$F_s = F(S, b'_{s-1}), \quad (4)$$

where F is a function of calculating the XOR frequency of the s^{th} bit segment;

for $s=1$, same as (S1.1);

(S3) calculating

$$P_s = P(S, b'_{s-1}), \quad (5)$$

where P is a function of calculating the start bit of a new bit segment;

for $s=1$, same as (S1.1);

(S4) stepping control variable for B bit segment in multi-frequency repeated XOR

$$b_s^f = P(S, b^{f-1}_s) \quad (6)$$

where P is a function of calculating the start bit of B bit segment for each frequency of the multi-frequency XOR;

b_s^f is the current frequency key bit segment (the same below);

$b_s^{f^*}$ is the random value determined by the current frequency (the same below);

for $S = 1$ and $f = 1$, same as (S1.1);

(S5) performing pseudo-random reconstruction to obtain:

(S5.1)

$$\underline{b_s^{fs}} = b_s^{fs}(P_s, L_s) \text{ 其中 } f_s = 1, 2, \dots, F_s; b_s^1(P_s, L_s) = b_s(P_s, L_s) \quad (7)$$

where b_s^{fs} is the F_s new bit segments of B determined by P_s, L_s, F_s, J_s based on the bit segment calculation unit in the previous step ;

DESCRIPTION

or

(S5.2)

$$\underline{b}^{fs}_s = \underline{b}^{fs}_s \left(\prod_{l=1}^{L_s} P_{s,j}(\underline{b}_j) \right) \quad (7')$$

where $\prod_{l=1}^{L_s} P_{s,j}(\underline{b}_j)$ represents the concatenation of bit segments of L_s bits obtained by hopping or no hopping determined by $J(b_{i's-1})$ after positioning with P ;

(S6) selecting $m_s = m_{s-1} + L_{s-1}$ by forward shifting and taking the stepping size of m_s same as the stepping size of b_s ,

(S6.1) performing XOR in the current step

$$c_s = m_s \oplus \underline{b}_s \quad (8)$$

or

(S6.2) performing multi-frequency XOR in the current step

$$c_s = (\dots (m_s \oplus \underline{b}_s^1) \oplus \underline{b}_s^2) \oplus \underline{b}_s^3 \dots \oplus \underline{b}_s^{F_s} \quad (8')$$

or

(S6.3) for (8'), swapping the paired bits selected with the pairing function C for c_s^f between XORs in each step

$$c_s = (\dots (c_s^f \oplus \underline{b}_s^2) \dots \oplus \underline{b}_s^{F_s}, c_s^f \text{ represents the target bit segment after swapping the matched bit pair of bit segments} \quad (8'')$$

and/or

(S6.4.1) for (8) / (8'), determining the increase value with the increase function A for c_s^f and b_s^f between XORs in each of steps, and increasing the bit segment by a subsequent bit of the plaintext, selecting the increased key bit in the same

DESCRIPTION

pseudo-random method as before, thereby increasing the XOR frequency by frequency

(S6.4.2) truncating bit segment of the current step cipher text in the length determined by L_s (same as (8''))

(S6.4.3) bitwise backfilling the increase of the accumulated increased XOR bit segment to the corresponding bit of plaintext

(S7) stepping

$$S=S+1 \quad (9).$$

The decryption processes of the present invention: (8), (8'), (8'') is changed to (8'''), (8''''), (8''''')

$$m_s = c_s \oplus \underline{b}_s \quad (8''')$$

$$m_s = (\dots (c_s \oplus \underline{b}_s^1) \oplus \underline{b}_s^2) \oplus \underline{b}_s^3 \dots \oplus \underline{b}_s^{F_s}) \quad (8''')$$

$m_s = (\dots (c_s^f \oplus \underline{b}_s^2) \dots \oplus \underline{b}_s^{F_s})$, where c_s^f represents the source bit segment after swapping the matched bit pair prior to XOR (8''''')

Note: (1) when there is a cipher text indirectly participating in a key run, such as in embodiments 4 and 5, the corresponding arguments of the pseudo-random reconstruction manipulation functions in decryption needs to be changed from the target random sequence to the source random sequence; (2) when swapping the paired bits between multiply frequencies are adopted, the swapping for target pairing after XOR processing is put forward to the swapping for source pairing before the XOR processing.

Description of the invention features

(1) $\underline{B}$ is derived from a pseudo-random reconstruction of different random bit

DESCRIPTION

segments of random sequences $B^1, B^2, \dots$, and is still a random sequence. Since the elements of the random sequences are independent of each other, the sequence keys obtained by the reconstruction for the random sequences depended on the variations in the position, the length and the frequency of the random elements are a random sequence as well. This target random key $\underline{B}$ depended on the seed random sequence B has concealment.

(2) The generation process of the random secret key is generated in a staggered manner accompanying the encryption process, that is: the position, the length and the next XOR frequency of key in the next step as well as the hopping control in selecting each key bit, the swapping control for target bit segment of each frequency and the increasing control for bit segment between frequencies are determined by the position function, the length determining function, the frequency determining function, the hopping function, the swapping function and the increasing function according to the value of the random number at the position of target sequence or associated position in the previous step, and the pseudo-random reconstruction process is accompanied throughout the encryption process.

(3) A nonlinear mixed morphology of the encryption process is presented based on XOR process with variable frequencies of orderly transitivity of XOR operation.

(4) The logic process of reconstructing the source random seed can be adjusted by the present methods. In particular, the reconstruction function is extended to a functional, and the chaos degree of the random element runs can be increased.

(5) The participation of the engine random number E and the feature quantity S enables each encryption process and each encrypted object having uniqueness. The change in S will improve the application space of the present invention:

DESCRIPTION

(5.1) in the encryption process, P, L, F, J, C and A allow the use of multiple $S = S_1, S_2, S_3, \dots, S_n$ to participate the pseudo-random reconstruction, and the value of bit segment or associated bit segment in the previous step is used to control whether each S_i participates the pseudo-random reconstruction or not;

(5.2) when S is a series of constants S^j , the encryption for a number of series of plaintext sequences using the sequence keys $\underline{B}^i$ respectively based on the same alternative random sequence B^J is supported, and when S is a number of series of variables generated by a plurality of functions, a plaintext sequence encryption architecture implemented with an exclusive sequence key $\underline{B}^f$ based on the same alternative random sequence B^J is supported.

(6) As one of the application selection, the alternative seed random sequences $B^1, B^2, \dots$ do not have to be generated during the encryption process (e.g., changed by a secret convention), so that the alternative seed random sequences can be previously stored at parties in the communication together with the manipulation functions P, L, F, J, C, A and S as a secret contract of the encryption and decryption parties. Such a contract can be altered and easy to be maintained.

(7) In practical applications, the plaintext can be prefixed with a piece of chaotic code to increase the difficulty of cracking.

(8) The pseudo-random reconstruction process in each step has the same computation amount, and the time complexity of the whole encryption process is $O((P+L+F+J)*n)$.

(9) Since P, L, F, J, C and A functions use the random value of key bit segment or associated bit segment in the previous step as argument for computing and randomly use a modulo operation, S is impossible to be back-stepped within the polynomial time complexity (other manipulations that need to be introduced also have this

DESCRIPTION

characteristic).

EMBODIMENTS

The invention will be further illustrated by the following examples.

Embodiment 0:

(S0.1) Length n of a random sequence is selected as 2048; one group of alternative random seed sequences are selected: $B^1: b^1_1, b^1_2, b^1_3, \dots, b^1_n$; for any determined number of engine $E1$, $P = (E+S) \% 2048$ is used to obtain a start position p of B^1 . B^1 is extracted starting from p by forward shifting and if b^1_n is reached the extraction continues from b^1_0 , thereby obtaining an instant random seed sequence $B = \{b^1_p, b^1_{p+1}, \dots, b^1_{p+2}, b^1_n, b^1_1, \dots, b^1_{p-1}\}$.

(S0.2) A unit length for calculating of each step is selected as 8 (1 byte), so (1) the corresponding maximum random value is 255; (2) the length of B is 256 bytes and a character array $B [256]$ with 256 elements can be made up.

(S1) It defines $L_s = 8$.

(S2) It defines $F_s = 1$.

(S3) It calculates stepping amount of a plaintext bit segment as $P(m_s) = P(m_{s-1}) + L_{s-1} = P(m_{s-1}) + 8$, and stepping amount of a key bit segment as $P(\underline{b}_s) = (P(\underline{b}_{s-1}) + L_{s-1}) \% 2048 = (P(\underline{b}_{s-1}) + 8) \% 2048$.

(S4) Stepping amount of a bit segment in B is 0 in a multi-frequency repeat XOR (because $F_s = 1$).

DESCRIPTION

(S5) $\underline{b}_s = b_s(P_s, L_s, F_s) = b_{s-1} + 8$.

(S6) XOR in the current step is performed $c_s = m_s \oplus \underline{b}_s$.

(S7) $S = S + 1$.

The decryption process is the same as the encryption process, but only (S6) is changed to (S6'):

(S6') $m_s = c_s \oplus \underline{b}_s$.

Embodiment 0 implements the simplest flow accompanied with pseudo-random reconstruction. Although practical significance is not large, the overall structure of the invention is shown.

Embodiment 1:

(S0.1) Length n of a random sequence is selected as 2048; two groups of alternative random seed sequences are selected: $B^1 : b^1_1, b^1_2, b^1_3, \dots, b^1_n$; $B^2 : b^2_1, b^2_2, b^2_3, \dots, b^2_n$; for any determined number of engines $E1$ and $E2$, $P = (E + S) \% 2048$ is used to obtain start positions p_1 and p_2 of B^1 and B^2 . B^1 and B^2 are paired to XOR by forward shifting starting from p_1 and p_2 respectively, thereby obtaining an instant random seed sequence $B = \{b^1_{p_1} \wedge b^2_{p_2}, b^1_{(p_1+1) \% 2048} \wedge b^2_{(p_2+1) \% 2048}, \dots, b^1_{(p_1+n) \% 2048} \wedge b^2_{(p_2+n) \% 2048}\} = \{b_1, b_2, b_3, \dots\}$.

(S0.2) A unit length for calculating of each step is selected as 8 (1 byte), so (1) the corresponding maximum random value is 255; (2) the length of B is 256 bytes and a character array $B [256]$ with 256 elements can be made up.

(S1) It defines L_s .

DESCRIPTION

(S1.1) When the remaining plaintext stepped to is greater than or equal to L_s , $L_s = B[P_{s-1} \% 256] \% 16 + 1$.

(S1.2) When the remaining plaintext stepped to is shorter than L_s , it sets $L_s =$ the remaining plaintext length.

(S2) It defines $F_s = B[P_{s-1}/8] \% 2 + 1$.

(S3) Stepping of a plaintext bit segment is $P(m_s) = P(m_{s-1}) + L_{s-1}$, and stepping of a key bit segment is $P(\underline{b}_s) = (P(\underline{b}_{s-1}) + L_{s-1}) \% 2048$.

(S4) Stepping of a bit segment in B is 0 in a multi-frequency repeat XOR is $P(\underline{b}_s^f) = (P(\underline{b}_{s-1}^{f-1}) + L_s) \% 2048$.

(S5) $\underline{b}_s = b_s(P_s, L_s)$ when $F_s = 1$; or $\underline{b}_s^{f(2)} = b^{f(2)}_s(P_s, L_s)$ when $F_s = 2$.

(S6) XOR in the current step is $C_s = m_s \oplus \underline{b}_s$ when $F_s = 1$;

or $C_s = (m_s \oplus \underline{b}_s) \oplus \underline{b}_s^{f(2)}$ when $F_s = 2$,

where the step size of m_s is taken same as that of $\underline{b}_s$.

(S7) $S = S + 1$.

The decryption process is the same as the encryption process, but only (S6) is changed to (S6'):

(S6') $m_s = c_s \oplus \underline{b}_s$ when $F_s = 1$;

or $C_s = (m_s \oplus \underline{b}_s) \oplus \underline{b}_s^{f(2)}$ when $F_s = 2$.

DESCRIPTION

$P(m_s)$, $P(\underline{b}_s)$ and $P(\underline{b}_s^f)$ are the start bits of plaintext bit in the current step, key bit in the current step and key bit at the current frequency, respectively.

% 2048 represents continuing from beginning when not only the last bit is reached.

Embodiment 1 implements a simple pseudo-random reconstruction process with variable length and variable frequencies by forward shifting.

Embodiment 2:

The following changes are made to Embodiment 1.

(S0.1²) Length n of a random sequence is selected as 524288; two groups of alternative random seed sequences are selected: $B^1 : b^1_1, b^1_2, b^1_3, \dots, b^1_n$; $B^2 : b^2_1, b^2_2, b^2_3, \dots, b^2_n$; for any determined number of engines $E1$ and $E2$, p_1 and p_2 are calculated with P being $p_s(B^i) = (p_{s-1}(B^i) * 12345 + 1103515245 + E_i + S) \% 524288$; a random seed sequence is calculated as $B = \{b_1, b_2, b_3, \dots\} = \{b^1_{p_1} \wedge b^2_{p_2}, b^1_{(p_1+1) \% 524288} \wedge b^2_{(p_2+1) \% 524288}, \dots, b^1_{(p_1+n) \% 524288} \wedge b^2_{(p_2+n) \% 524288}\}$.

(S0.2²) A unit length for calculating of each step is selected as 16 (2 bytes), so (1) the corresponding maximum random value is 65535; (2) the length of B is 655356 bytes and a character array B [66536] with 65536 elements can be made up.

(S1.1²) It defines $L_s = B[P_{s-1} \% 65536] \% 68 + 64$ when the remaining plaintext stepped to is greater than or equal to L_s .

(S1.2²) When the remaining plaintext stepped to is less than L_s , it sets $L_s =$ the remaining plaintext length (in decryption, L_s may not be changed, but instead the redundant plaintext is discarded after the calculation is complete).

DESCRIPTION

(S2²) It defines $F_s = (S + B[P_{s-1} \% 65536]) \% 8 + 16$.

(S3²) The position for B in the embodiment (S3) is changed as

$$P(\underline{b}_s) = (B[(P(\underline{b}_{s-1}) + S) \% 65536] + 1103515245) \% 524288.$$

(S4²) The position for B in the embodiment (S4) is changed as

$$P(\underline{b}_s^f) = (B[(P(\underline{b}_{s-1}^{f-1}) + S) \% 65536] + 1103515245) \% 524288,$$

$$f=1,2,\dots,F_s; P(\underline{b}_s^{f(1)})=P(\underline{b}_s).$$

By above modifications, (1) the XOR frequency of multi-frequency XOR is increased; (2) the chaos degree of selecting a corresponding bit segment is improved by changing the selection mode for bit segment of the key B from a forward shifting mode to a pseudo-random mode of determining the position based on a value in the previous step; (3) the modulus in (S1.1²) is 68 rather than 64 or 32, which allows the longest length for selecting bit segment to be more than twice of the shortest length.

The embodiment 2 implements the encryption process in which key in the current step is pseudo-randomly reconstructed by the random value of the key bit segment in the previous step.

Embodiment 3:

The following changes are made to Embodiment 2.

(S0.4³) A further random sequence B^T is defined with length equal to that of B.

(S2³) It defines $F_s = (S + B[\underline{b}'_{s-1} \% 65536]) \% 8 + 16$.

(S3³) The positioning for B in the embodiment (S3²) is changed to position for B^T :

DESCRIPTION

$$P(\underline{b}_s^T) = (B[(\underline{b}'_{s-1} + S) \% 65536] + 1103515245) \% 524288.$$

(S4³) The positioning for $\underline{B}$ in the embodiment (S4²) is changed as

$$P(\underline{b}_s^{T,f}) = (B[(\underline{b}'^{f-1}_s + S) \% 65536] + 1103515245) \% 524288;$$

$$f=1,2,\dots,F_s; P(\underline{b}_s^{T,f(1)}) = P(\underline{b}_s^T).$$

By above modifications, the random value obtained from $\underline{B}$ to perform pseudo-random reconstruction for $\underline{B}^T$ also allows $\underline{B}$ having concealment and chaotic characteristics.

Embodiment 3 implements the process of scheduling another random sequence from a random value obtained from the previous step to implement a pseudo-random reconstruction for key.

Embodiment 4:

The following changes are made to Embodiment 2.

(S0.4⁴) It defines the bit length to which the cipher text $\underline{C}$ is reached in the current step is c_s .

$$(S1.1^4) \text{ The length of bit segment is modified: } L_s = B[P_{s-1} \% 65536] \% (c_{s-1}/8) + 64.$$

By above modifications, an XOR bit segment of $\underline{B}$ is selected by utilizing the progress of the generated cipher text without changing the concealment and chaotic characteristics of $\underline{B}$.

Embodiment 4 implements the process of controlling the pseudo-random reconstruction for key according to the progress of the cipher text sequence.

DESCRIPTION

Embodiment 5:

The following changes are made to Embodiment 2.

(S0.2⁵) A unit length for calculating of each step is selected as 20 (2.5 bytes), so the corresponding maximum random value is much greater than 65535; it is modulo 65536, and still distributed in character array B [65536] in byte.

(S7⁵) The positioning function for B in (S3²) is selected by the formula $i=(b'_{s-1}*S)\%3$ from the following three functions:

$$P[0](b_s) = (B[(b'_{s-1}+S)\% 65536] \oplus (B[c'_{s-1}\%65536]))\%524288;$$

$$P[1](b_s) = (B[b'_{s-1}/8] + (S*12347))\%524288;$$

$$P[2](b_s) = (B[(b'_{s-1}+S+1103515245)\% 65536])\%524288;$$

where c'_{s-1} represents the value taken from a cipher text in the previous step in the agreed calculation unit, and the chaos degree of the key element run is further improved.

Note that (1) the decryption algorithm in this embodiment is modified, such that c'_{s-1} is the generated term of the corresponding construction function in encryption while c'_{s-1} is the source item of the corresponding construction function in decryption; (2) the cipher text bit segment is not used to construct the key bit segment, but instead is used to change the run of the key construction.

Embodiment 5 implements a run of plaintext indirectly scheduling pseudo-random reconstruction for key (note that the unit length for calculation is 20 bits).

Embodiment 6:

The following changes are made to Embodiment 2.

DESCRIPTION

(S0.4⁶) It defines an array $S[2] = \{\text{cell number of a user; a user name;}\}$.

(S0.5⁶) It defines $K = S[\underline{b}_{s-1} \% 2]$.

(S2⁶) It defines $F_s = (S[\underline{b}'_{s-1} \% 2] + B[P_{s-1}/8]) \% 8 + 16$.

(S3⁶) The positioning for B in (S3²) is changed as

$$P(\underline{b}_s) = (B[(\underline{b}'_{s-1} + (S[\underline{b}'_{s-1} \% 2])) \% 65536]) \% 52428.$$

(S4⁶) The positioning for B in (S4²) is changed as

$$P(\underline{b}_s^f) = (B[(\underline{b}'_{s-1}^{f-1} + (S[\underline{b}'_{s-1} \% 2])) \% 65536]) \% 524288, \quad f=1, 2, \dots, F_s.$$

The chaos degree of the run of positioning and determining frequency is further improved.

Embodiment 6 implements the process of implementing the pseudo-random reconstruction for key from a random value obtained from the previous step and a plurality of feature values.

Embodiment 7:

The following changes are made to Embodiment 2.

(S0.1⁷) One group of alternative random seed sequences is added: B^3 : $b^3_1, b^3_2, b^3_3, \dots, b^3_n$. Another seed random sequence B_2 is generated from B^2 and B^3 using the same method.

DESCRIPTION

(S0.4⁷) It defines a seed sequence array $B[2] = \{B_1, B_2\}$.

(S0.5⁷) It defines $G = B[\underline{b}_{s-1}] \% 2 + 1$.

(S2⁷) It define $F_s = (S + B_G[P_{s-1}/8]) \% 8 + 16$.

The following changes are made to Embodiment 6.

(S3⁷) The positioning for B in (S3²) is changed as $P(\underline{b}_s) = (B_G[(\underline{b}'_{s-1} + (S[\underline{b}'_{s-1}] \% 2))) \% 65536] \% 524288$.

(S4⁷) The positioning for B in (S4²) is changed as $P(\underline{b}_s^f) = (B_G[(\underline{b}'^{f-1}_s + (S[\underline{b}'^{f-1}_s] \% 2))) \% 65536] \% 524288$.

The chaos degree of the run of positioning and determining frequency is further improved.

Embodiment 7 implements the process of pseudo-randomly selecting the seed random sequence reconstructed key in the current step from a plurality of seed random sequences.

Embodiment 8:

The following changes are made to Embodiment 2.

(S0.1⁸) One group of alternative random seed sequences is added: B^3 : $b^3_1, b^3_2, b^3_3, \dots, b^3_n$. Another seed random sequence B_2 is generated from B^2 and B^3 using the same method.

(S0.4⁸) It defines a seed sequence array $B[2] = \{B_1, B_2\}$.

DESCRIPTION

(S0.5⁸) It defines $G=B[\underline{b}_{s-1}\%2]+1$.

(S2⁸) It define $F_s = (K+B_G[P_{s-1}/8])\%8+16$.

(S3⁸) The positioning for B in (S3²) is changed as $P(\underline{b}_s)=(B_1[\underline{b}'_{s-1}\%65536])\oplus (B_2[\underline{b}'_{s-1}\%65536])$.

(S4⁸) The positioning for B in (S4²) is changed as $P(\underline{b}_s^f)=(B_1[\underline{b}'_{s-1}^{f-1}\%65536])\oplus (B_2[\underline{b}'_{s-1}^{f-1}\%65536])$.

The chaos degree of the run of positioning and determining frequency is further improved.

Embodiment 8 implements the process of implementing pseudo-random reconstruction for key using a plurality of seed random sequences at the same time.

Embodiment 9:

The following changes are made to Embodiment 2.

(S0.1⁹) One group of alternative random seed sequences is added: B^3 : $b^3_1, b^3_2, b^3_3, \dots, b^3_n$. Another seed random sequence B_2 is generated from B^2 and B^3 using the same method.

(S0.4⁹) It defines a seed sequence array $B[2]=\{B_1, B_2\}$.

(S0.5⁹) It defines $G=B[\underline{b}'_{s-1}\%2]$.

(S2⁹) It define $F_s = (B_G[\underline{b}'_{s-1}/8])\%8+8$.

DESCRIPTION

(S3⁹) The positioning for B in (S3²) is changed as positioning for two seeds respectively $P(\underline{b}_{1,s})=B_1[(\underline{b}'_{s-1}+S)\%65536]$, $P(\underline{b}_{2,s})=B_2[(\underline{b}'_{s-1}+S)\%65536]$.

(S4⁹) The positioning for B in (S4²) is changed as positioning for two seeds respectively $P(\underline{b}_{1,s}^f)=B_1[(\underline{b}'_{s-1}^{f-1}+S)\%65536]$, $P(\underline{b}_{2,s}^f)=B_2[(\underline{b}'_{s-1}^{f-1}+S)\%65536]$;

$$f=0,1,\dots,F_s.$$

The encryption process is defined as:

$$(S6^9) \text{ in the current step, } c_s = (\dots(m_s \oplus \underline{b}_{1,s}) \oplus \underline{b}_{2,s}) \oplus \underline{b}_{1,s}^{f_2} \oplus \underline{b}_{2,s}^{f_2} \dots \oplus \underline{b}_{1,s}^{F_s} \oplus \underline{b}_{2,s}^{F_s}.$$

The chaos degree of the run of positioning and determining frequency is further improved.

The decryption process is the same as the encryption process, but only (S6⁹) is changed to (S6^{9'}):

$$(S6^{9'}) m_s = (\dots(c_s \oplus \underline{b}_{1,s}) \oplus \underline{b}_{2,s}) \oplus \underline{b}_{1,s}^{f_2} \oplus \underline{b}_{2,s}^{f_2} \dots \oplus \underline{b}_{1,s}^{F_s} \oplus \underline{b}_{2,s}^{F_s}.$$

Embodiment 9 implements the process of scheduling multiple seed random sequences from a random value obtained from the previous step to directly perform multi-frequency XOR encryption on plaintext.

Embodiment 10:

The following changes are made to Embodiment 9.

(S5¹⁰) It implements key pseudo-random reconstruction with or without hopping.

(S5.1¹⁰) It defines $J^1 = \{j_{1,1}^1, j_{1,2}^1, \dots, j_{1,L}^1\}$, where $j_{1,1}^1$ is the first bit of $B_1[\underline{b}'_{s-1}]$ and the subsequence shifts forward in turn;

DESCRIPTION

$J^2 = \{j^2_1, j^2_2, \dots, j^2_L\}$, where j^2_1 is the first bit of $B_2[b'_{s-1}]$ and the subsequence shifts forward in turn;

$J^{f,1} = \{j^{f,1}_1, j^{f,1}_2, \dots, j^{f,1}_L\}$, where $j^{f,1}_1$ is the first bit of $B_1[b'^{f-1}_s]$ and the subsequence shifts forward in turn;

$J^{f,2} = \{j^{f,2}_1, j^{f,2}_2, \dots, j^{f,2}_L\}$, where $j^{f,2}_1$ is the first bit of $B_2[b'^{f-1}_s]$ and the subsequence shifts forward in turn;

$f=1, 2, \dots, F_s$ (note that the argument domain of J is not $B_1[(P(b_{s-1})+16)\%65536]$).

(S5.2¹⁰)

Each bit is selected from B_i by the following formula according to the element in above array J hopping forward or not hopping to concatenate the key bit segment:

$$\boxed{b_{i,s}} = \prod_{l=1}^{L_S} b_{i,l} > j_l \quad , \quad \boxed{b_{i,s}} = \prod_{l=1}^{L_S} b_{i,l} > j_l$$

Embodiment 10 implements the selection of a plurality of key segment bits with or without hopping to perform a multi-frequency encryption process on a plaintext bit segment directly.

Embodiment 11: (S6⁹) in Embodiment 9 is decomposed as following:

(S0.6¹¹) establishing a pairing function C^f for determining that the bits to be swapped are $L/2+8$ and $L/2-8$ from the start bit in the current step;

(S6.1¹¹) decomposing (S6⁹) as F_s steps:

$$c_1 = m_s \oplus b_{1,s}; \quad c_2 = c_1 \oplus b_{2,s}; \quad c_3 = c_2 \oplus b_{1,s}^{f_2}; \quad \dots; \quad c_{F_s} = c_{F_s-1} \oplus b_{2,s}^{F_s};$$

(S6.2¹¹) pairing the target bit segment c_{fs} for swapping after XOR of each step:

$$c_{fL/2+8} \Leftrightarrow c_{fL/2-8}.$$

DESCRIPTION

Note that: in the decryption, the above pairing for swapping is performed on the source bit segment (cipher text) prior to XOR in each step of (S6.1¹¹).

Embodiment 11 implements an encryption process with pseudo-random reconstruction with bit pairing for swapping.

Embodiment 12:

(S6.2.0¹²) The bit length in (S6.12¹¹) in embodiment 11 is extended by increase amount calculated by A in each frequency

$$A_{f,s} = L_{f-1,s} + (S * F_{s+12345}) \% 3 + 1.$$

(S6.2.1¹²) For the target of XOR in each frequency, the extended bits from a plaintext are concatenated until the plaintext reaches the end (in decryption, the extended bits from cipher text are concatenated)

$$C_f = (C_{f-1} \oplus \underline{b_{2,s}}) || m_{(L_{f-1,s} + S \% F_s)}; \dots; c_{F_s-1} = (c_{F_s-2} \oplus \underline{b_{1,s}}) || m_{(L_{F_s-1,s} + S \% F_s)}.$$

(S6.2.2¹²) The extended portion is not taken from the cipher text in the Sth step, but instead all bits of the extended portion are backfilled to the corresponding bits of the plaintext to participate in XOR in the next step.

The decryption process is the same as the encryption process, but only (S6.2¹²) is changed to (S6.2^{12'}):

(S6.2^{12'}) for the target of XOR in each frequency, the extended bits from the cipher text are concatenated until the cipher text reaches the end

$$m_f = (m_{f-1} \oplus \underline{b_{2,s}}) || c_{(L_{f-1,s} + S \% F_s)}; \dots; m_{F_s-1} = (m_{F_s-2} \oplus \underline{b_{1,s}}) || c_{(L_{F_s-1,s} + S \% F_s)}.$$

(S6.2.2^{12'}) The extended portion is not taken from the plaintext in the Sth step, but instead all bits of the extended portion are backfilled to the corresponding bits of the cipher text to participate in XOR in the next step.

DESCRIPTION

Note that: the source bit segment in the current step has been partially processed by the S^{th} step, that is, the starting portion of the target bit segment in the $S+1^{\text{th}}$ step is XORed with both the key in the S^{th} step and the key in the $S+1^{\text{th}}$ step. Since each step follows a pseudo-random convention at work, the process does not change in the decryption, and the plaintext can be recovered by the decryption.

Embodiment 12 implements a variation of the length of bit segment in XOR of each frequency. This makes the accompanied pseudo-random reconstruction method more prone to the effect of one chaotic codebook per time.

Embodiment 13:

The following changes are made to Embodiment 9.

(S0.1¹³) Length n of a random sequence is selected as 524288; two groups of alternative random seed sequences are selected: $B^1 : b^1_1, b^1_2, b^1_3, \dots, b^1_n$; $B^2 : b^2_1, b^2_2, b^2_3, \dots, b^2_n$; for any determined number of engines $E1$ and $E2$, p_1 and p_2 are calculated with P being $p_i = ((p_{i-1}+1)*12345+1103515245+E+S)\%524288$.

(S0.3¹³) It defines $F^j_s = (S + B[P^j_{s-1}\%65536])\%8+16$.

(S0.4¹³) It defines a positioning function for B^j $P(b^j_s) = B[(b^j_{s-1}+S)\%65536]\%524288$.

(S0.5¹³) It defines the length L_s of bit segment of the seed random sequence for pseudo-random reconstruction.

(S0.5.1¹³) When the remaining length of the alternative random sequence B^i is greater than or equal to L_s , $L_s = B^1[b^1_{s-1}\%65536]\%16+64$.

DESCRIPTION

(S0.5.2¹³) When the remaining length of the alternative random sequence B^i is less than L_s , it sets L_s = the remaining plaintext length.

$$(S0.6^{13}) \underline{b}_s^j = b_s^j(P_s^j, L_s^j, F_s^j).$$

$$(S0.7^{13}) \text{ XOR in the current step } b_s = b_s^1 \oplus b_s^2.$$

$$(S0.8^{13}) S=S+1.$$

Then the chaos degree of obtaining the seed random sequence is improved.

Embodiment 13 implements a pseudo-random reconstruction process with a seed random sequence in a manner of non-forward shifting.

The various manipulation functions used in the various embodiments described above all have a large adjustable space, and provide sufficient application space for the method accompanied with pseudo-random construction.

CLAIMS

1. A sequence encryption method accompanied with adjustable randomly-reconstructed key, characterized by sequence encryption/decryption accompanied with reconstructing a random sequence key based on a random binary sequence during encryption, the method comprising steps of:

(1) dynamically constructing a seed random sequence B_i at beginning of each encryption through an existing random sequence, wherein the seed random sequence B_i is derived from a pseudo-random reconstruction of an alternative random sequence B^j , or is directly taken from the alternative random sequence B^j ; wherein B^j is a part of a contract of encryption and decryption parties, and the seed random sequence B_i is pseudo-randomly reconstructed or truncated from B^j starting from an agreed position according to an agreement;

(2) pseudo-randomly constructing a random key $\underline{B}$ using the seed random sequence B_i with a bit segment of a fixed or unfixed length; wherein the generation of the random key $\underline{B}$ is accompanied during encryption/decryption process and the random key $\underline{B}$ is generated step by step through pseudo-randomly reconstructing each bit in the seed random sequence B_i ; selecting a key bit segment for XOR in each step according the secret agreement of the encryption and decryption parties, wherein the key bit segment comprises frequencies of XOR on a plaintext, a starting position of bit segment at each frequency and length of the bit segment, and a process of the pseudo-random reconstruction is determined by the secret agreement; and

(3) encrypting/decrypting a segment of the plaintext by the bit segment of the fixed or unfixed length, wherein each segment of the plaintext XORs with several corresponding key bit segments obtained by the pseudo-random reconstruction with a fixed frequency or variable frequency, an increase or no increase between frequencies and swapping or not swapping a cipher text between frequencies, to generate the cipher text.

2. The sequence XOR method accompanied with adjustable randomly-reconstructed key according to claim 1, characterized by the secret

CLAIMS

convention being implemented by an agreement construction function which uses following random values as arguments: ① a random value determined by the key bit segment in previous step according to the secret agreement for the pseudo-random reconstruction, ② a random values determined by a pseudo-randomly agreed bit segment of other random sequences according to the secret agreement for the pseudo-random reconstruction, or ③ a random value determined by a value of a cipher text bit segment in the previous step according to the secret agreement for the pseudo-random reconstruction.

3. The sequence XOR method accompanied with adjustable randomly-reconstructed key according to claim 1, characterized by:

(1) forward shifting the bit segment of the plaintext in turn in the process of encryption/decryption for the plaintext segment by segment, wherein the length of the bit segment of the plaintext is depended on the length of the key bit segment, and the length l of the key bit segment $\underline{b_s}$ and the starting position p of the seed random sequence B_i are determined by the contract agreed by the encryption and decryption parties; a same bit segment of the plaintext can be subjected to multi-frequency XOR in turn by a plurality of key bit segments, and the frequency f of the multi-frequency XOR is determined by the contract agreed by the encryption and decryption parties;

(2) determining that the position p of the key bit segment $\underline{b_s}$ in the current step is the position by forward or backward shifting the position of the key bit segment $\underline{b_{s-1}}$ in the previous step, that is, backward or forward shifting several bits according to the position of the bit segment in the previous step, wherein the position p of the key $\underline{b_s}$ is forward or backward shifted to reach the last bit or first bit of the seed random sequence B , the forward or backward shifting continues from the last bit or the first bit;

(3) determining that the length l of the key bit segment $\underline{b_s}$ in the current step is the length of the bit segment $\underline{b_s}$ in the current step selected as the length of the key bit segment $\underline{b_{s-1}}$ in the previous step;

(4) determining that the frequency f of XOR in the current step is a frequency

CLAIMS

taken as a positive integer, wherein the pseudo-random reconstruction process of the seed random sequence B presents a morphology of forward or backward shifting equidistant from the plaintext when the selected frequency is 1, and the pseudo-random reconstruction process of the seed random sequence B presents a morphology of forward or backward shifting with hopping in times of distance from the plaintext when the selected frequency is greater than 1; and

(5) encrypting the bit segment of the plaintext with XOR at the fixed frequency step by step using the key bit segment $\underline{b}_s$ obtained above.

4. The sequence XOR method accompanied with adjustable randomly-reconstructed key according to claims 1 or 3, characterized by improving encryption density by following steps:

(1) determining a unit for calculating the value of the bit segment in the current step of the pseudo-random reconstruction, wherein the unit for calculating is determined by the agreed starting position and agreed length of the bit segment $\underline{b}_{s-1}$ in the previous step;

(2) determining the starting position p_s of the bit segment $\underline{b}_s$ of the random key $\underline{B}$ in the current step by a random value of random number taken in the unit for calculating of the current time of the bit segment $\underline{b}_{s-1}$ or an associated bit segment of the random key $\underline{B}$ in the previous step and a feature quantity S through a positioning function P , wherein P is any function whose value domain covers the whole seed random sequence B , that is, the value domain $P(\underline{b}_s)$ of P satisfies $0 < P(\underline{b}_s) \leq n$, and n is the maximum position of element in B ;

(3) determining the length l of the bit segment $\underline{b}_s$ of the random key $\underline{B}$ in the current step by the value of the random number of the bit segment $\underline{b}_{s-1}$ or an associated bit segment of the random key $\underline{B}$ in the previous step and the feature quantity S through a length determining function L , wherein L can be any function whose value domain is greater than 0, that is, the value domain $L(\underline{b}_s)$ of L satisfies $0 < L(\underline{b}_s) \leq n$, and n is the maximum position of element in B ;

(4) determining the frequency f of XOR for the bit segment m_s of the plaintext

CLAIMS

with the random key bit segment b_s by the random value of the bit segment $\underline{b_{s-1}}$ or an associated bit segment of the random key $\underline{B}$ in the previous step and the feature quantity S through a frequency determining function F , wherein F is any function whose value domain is greater than 0;

(5) determining the key bit segment b_s in the seed random sequence B by substituting p and/or l selected by the steps (2) and (3) for previous p and/or l , determining the frequency f of XOR in the current step by the step (4), when f is greater than 1, selecting f bit segments from B in a manner of forward shifting or selecting f bit segments $b_s, b_{s2}, \dots, b_{sf}$ by randomly positioning in B in the manner of the step (2), and performing the multi-frequency XOR f times orderly for the bit segment m_s of the plaintext in the current step with the above selected key bit segments.

5. The sequence XOR method accompanied with adjustable randomly-reconstructed key according to claims 1 or 3 or 4, characterized by further improving encryption density by following steps:

(1) pseudo-randomly constructing the random key $\underline{B}$ with a plurality of seed random sequences B_i ($i = 1, 2, 3, \dots, m$; m is any positive integer), wherein the constructing comprises: (a) pseudo-randomly selecting a plurality of $b_{i,s}$ according to the value of the key bit segment $\underline{b_{s-1}}$ or the associated bit segment in the previous step in an agreed process; (b) pseudo-randomly selecting one of $b_{i,s}$ as the key bit segment $\underline{b_s}$ in the current step according to the value of the key bit segment $\underline{b_{s-1}}$ or the associated bit segment in the previous step in the agreed process; or (c) after orderly paring for $b_{i,s}$ ($i = 1, 2, 3, \dots, m$) one by one to XOR, selecting the generated result as the key bit segment $\underline{b_s}$ in the current step;

(2) performing f -frequency ordered XOR on the bit segment m_s of the plaintext through the key bit segment $\underline{b_s}$ constructed by above (1).(b) or (1).(c) and $\underline{b_s^f}$ constructed in same way; or

(3) directly using various sub bit segments $b_{i,s}$ constructed by above (1).(a) as m key bit segments, selecting $\underline{b_{i,s}^f}$ one by one by any of above P , L and F functions, and then performing the multi-frequency XOR with $m*f$ on the bit segment m_s of the

CLAIMS

plaintext one by one or in a mixing way.

6. The sequence XOR method accompanied with adjustable randomly-reconstructed key according to claims 1 or 3 or 4 or 5, characterized by further improving encryption density by following steps:

(1) constructing a bit hopping function J for selecting the key bit segment in the current step;

(2) in obtaining $\underline{b}_{i,s}$ and/or $\underline{b}_{i,s}^f$, extracting each bit of each key bit segment in the current step or at the current frequency bit by bit starting from a source random sequence position determined by P with or without hopping according to the hopping function J , until reaching the length l ; and

(3) performing the multi-frequency ordered XOR for the bit segment m_s of the plaintext in the manner of above 5.(2) or 5.(3).

7. The sequence XOR method accompanied with adjustable randomly-reconstructed key according to claims 1 or 3 or 4 or 5 or 6, characterized by adding a step of pairing bits of the target bit segment between frequencies of the multi-frequency XOR to swap according to a pseudo-random secret agreement in the process of multi-frequency XOR:

(1) constructing a bit pairing function C for selecting the target bit segment in the current step;

(2) performing swapping for the target bit segment of each frequency with the selected paired bits during encrypting in the manner of above 5.(2) or 5.(3); and

(3) performing swapping for the source bit segment (cipher text) with the selected paired bits during decrypting in an inverse manner of above 5.(2) or 5.(3).

8. The sequence XOR method accompanied with adjustable randomly-reconstructed key according to claims 1 or 3 or 4 or 5 or 6 or 7, characterized by adding a step of increasing the length of bit segment for XOR per frequency according to the pseudo-random secret agreement in the process of

CLAIMS

multi-frequency XOR:

(1) constructing a length extension function A for the bit segment between frequencies in the current step;

(2) extending the source bit segment (plaintext) and the key bit segment by a determined increase value frequency by frequency during encrypting in the manner of above 5.(2) or 5.(3); and

(3) extending the source bit segment (cipher text) and the key bit segment by the determined increase value frequency by frequency during decrypting in the manner of above 5.(2) or 5.(3);

(4) still selecting a length of bit segment without increase as the length of the target bit segment in the current step;

(5) backfilling the accumulated increased bits of the bit segment in calculating to the corresponding positions of the source random sequence.

9. The sequence XOR method accompanied with adjustable randomly-reconstructed key according to claims 1 or 3 or 4 or 5 or 6, characterized by adding an alternative random sequence and implementing a step of pseudo-randomly constructing the seed random sequence to improve a chaos degree of the seed random sequence B:

(1) pre-generating more than one alternative random sequence B^j for constructing the seed random sequence B, establishing a positioning function P_0 and the feature quantity S by using an instant random number E^j called as engine the number of which matches B^j , and after the starting position of each B^j is determined by P_0^j driven with E^j and S, obtaining b_s by ordered XOR with b_s^j and b^{j-1}_s segment by segment, wherein in the process of XOR, it continues with the bit of any B^j from the starting of B^j when the last bit of the B^j is reached, until reaching a desired length of the seed random sequence B;

(2) determining the corresponding bit segment of the seed random sequence B from the bit segment of b^{j-1}_s or b_{s-1} by substituting P, L, F, J, C and A methods for corresponding methods for generating a pseudo-random reconstruction element in (1),

CLAIMS

constructing the seed random sequence B segment by segment, and changing the length of the generated seed random sequence B as desired; and

(3) for a plurality of seed random sequences B_i , repeating the above steps (1) and (2) to obtain different B_i .

10. The sequence XOR method accompanied with adjustable randomly-reconstructed key according to claims 4 or 5 or 6 or 7 or 8 or 9, characterized by: the feature quantity S participating in the pseudo-random reconstruction being a constant or a variable of a function, a plurality of constants or variables of the function or an integration of the plurality of constants or variables of the function;

(1) in the encryption process, P, L, F, J, C and A allowing to use a plurality of $S = S_1, S_2, S_3, \dots, S_m$ to participate in the pseudo-random reconstruction, and pseudo-randomly determining whether each S_i is allowed to participate in the pseudo-random reconstruction by a manipulation function K through any of above manners, that is, randomly selecting k S_i from the m S_i to participate in calculating, $k \leq m$; and

(2) when S is a plurality of groups of feature quantities S^j for the pseudo-random reconstruction, supporting the sequence encryption for j series of plaintext based on j respective exclusive sequence keys $\underline{B}^j$ of the common seed random sequence B_i , and when S is a plurality of groups of variables generated by a plurality of functions, supporting a plaintext sequence encryption architecture implemented based on respective exclusive sequence keys $\underline{B}^j$ of the common seed random sequence $\underline{B}_i (i = 1, 2, 3, \dots, k)$.