A Lightweight Ciphering Method Inspired by Spreading Activation Theory

Xixu Fu, *Member, IAENG*, Hu Li

***Abstract*—A spreading activation trajectory ciphering method inspired by the spreading activation theory in psychology field is proposed. Using the plain text as the activation source, a spreading activation trajectory is generated as cipher text by two actions which are called spread and activation. The spreading operation finds a point suit for the representation of a character near the previous point. The activation operation finds the position of next point. The activation operation can improve the utilization of cipher space to gain low mutual information and coefficient. The method is simple and flexible. It consumes almost no more time when its key becomes much longer. As a result, it is almost immune to brutal enumerating attacks. Low mutual information and low coefficients between plaintext and cipher text can prevent the method from statistical attack.**

***Index Terms*—cryptography, security, spreading activation, trajectory, cipher space.**

I. Introduction

CREATE a safe encryption system is one of the most important issues in cryptography. The essence of safety is to avoid attackers to get the plaintext while enable the plaintext legible to users with passwords. There are two important aspects to ensure safety. One is to prolong the time consumed in deciphering, and the other is to protect the message from attacks other than enumerating. The two aspects are both important in the design of an encryption system. However, the former aspect becomes harder and harder because of the development of powerful machines. It is also hard to generate a system which can avoid attacks such as differential analysis [17]. Fortunately, the study of psychology can provide a potential solution.

Nature is a complex system which is hard to be represented and enumerated even by the most powerful machine. At the beginning the way used to represent knowledge is simple. Researchers tried to get knowledge by search many years ago [18]. After that, complicated knowledge systems such as semantic dictionaries have been implemented to simulate the knowledge background of human [7][15][24]. However, these systems are far from the detailed human knowledge structure. Spreading activation theory was advanced to explain the human reason process in nature [6]. The theory is good at creating natural scenarios from human knowledge. People can recall a scenario by repeating the activation trajectory generated in the scenario. Without the principle and background knowledge, it is impossible to reach a correct result. It is a good idea to use the background knowledge as the key and spreading activation process as the encryption process.

Inspired by the spreading activation process, an encryption method is proposed. In this method a space with values is regarded as knowledge background, plaintext as the stimulation source, and the output activation trajectory as cipher text. Then the efficiency and security of this algorithm are analyzed and compared with other methods.

Manuscript received Nov. 17, 2022; revised Feb. 8, 2023.

Xixu Fu is a Lecturer of Shanghai Ocean University, Shanghai 201306, China (e-mail: xxfu@shou.edu.cn).

Hu Li is a Lecturer of Shanghai Lixin University of Accounting and Finance, China (corresponding author e-mail: fishfxx@163.com).

II. Motivation and Related Work

A. Spreading activation theory

Spreading activation theory was proposed by Collins A. M. in 1975 in the study of memory and language [3]. Anderson J R introduced a complete theory on spreading activation process and its architecture [1]. The theory proposes that the semantic networks that represent specific semantic memories are organized into a larger network that comprises concepts. The relation of the concepts can be strengthened in a learning process called activation. The theory has been widely used in text semantic understanding [2][5][9][26] and recommendation systems [4]. Related software has also been developed to simulate the spreading and activation process [23]. Fu's paper described the application of spreading activation theory in the field of scenario reasoning [6].

Spreading activation reasoning is the process of continuously activating human knowledge and constructing a conceptual system according to the acquired information. For example, when the concept "salt" is acquired, we may think of concepts related to salt such as "salty" and "soluble". Then, when the next concept "water" is acquired, our attention may focus on the concept of "soluble" and finally we form a certain scenario. The formation process of this scenario is closely linked. Different knowledge systems and inputs including different input sequences will affect the final scenario and the corresponding activation trajectory.

Although the spreading activation theory is mainly applied in the fields of conceptual and semantics reasoning, it can be a good reference for cryptography. The correspondence of input concepts and activation trajectory is just like the correspondence of plaintext and cipher text. The complex and unpredictable actions in the cipher space can provide security.

Let's take a well-known image as a cipher space. Figure 1 shows a trajectory found for the sentence "Hello world!" by

fitting ASCII code of characters to the color of pixels.

Fig. 1. A trajectory for “Hello world!” in the picture Lenna.

B. Evaluation methods

The first criterion for an encryption method is strength, which means the time consumed to enumerate possible passwords. However, the strength is not enough for a good encryption method. To defend differential attack and statistics-based attacks, balancedness and nonlinearity are other important metrics for encryption methods [8][10]. S-box is often used to reduce linearity [20]. Some functions are developed to gain balancedness [8].

Entropy is a popular metric of information [22]. Mutual information can be used to measure the relation of two discrete variables such as plaintext and cipher text. It can be defined as follows:

$$I(C,T)=\sum_{c\in C}\sum_{t\in T}p(c,t)\log\frac{p(c,t)}{p(c)p(t)} \quad (1)$$

C and T represent two variables which mean cipher text and plaintext respectively. Their possible values are represented as c and t in the formula. Mutual information varies from 0 to 1. Higher value means closer relation.

Correlation coefficient is a metric to evaluate the relation between two groups of numbers [13]. Although the metric can not be used to measure the relationship between the characters and their whole representations directly, it can be used to measure the relation between the ASCII code of characters and each component of their representations.

C. Related encryption methods

1) Simple substitution methods

Base64 is a popular block encryption algorithm [19]. The algorithm uses 24 bits (3 characters) as a block to encrypt. The same plaintext block corresponds to the same cipher text. It is easy to know that if all possible cipher text corresponding to the blocks that can be enumerated, any cipher text can be easily decrypted. For base64 encryption, the number of calculations required is 224 or 2563. If the encryption object is limited to English characters that can be displayed, the amount of calculation will be smaller.

2) DES and AES

DES is another block encryption algorithm [12]. The algorithm uses a 56-bit password block to encrypt a 64-bit plaintext block. Because of the short password, the strength of the algorithm is weak [10]. The algorithm is also vulnerable to side channel attack [12].

AES is an advanced standard of encryption [11]. The algorithm uses bigger blocks and longer passwords. The strength of AES is better than that of DES. AES has many modes, and CBC is a popular one. There are also many methods to attack AES systems [14][16][20][27].

3) Stream cipher RC4

RC4 is a common stream encryption algorithm [21]. It is famous about the low correlation between its plaintext and cipher text. The algorithm steps are as follows [13]:

- Use n numerical values (usually 0-255) to generate a random array s according to the ASCII value of the password by exchange operation.
- According to the plaintext length l and S, the encryption string S_k with length l is generated by the exchange operation.
- Use elements in plaintext to XOR the corresponding elements in S_k to generate cipher text.

The algorithm has good balance and nonlinearity. Because step 2 can almost guarantee that the cipher text of the same characters in different positions is not the same, it is strong for brute force attack. However, when the password is determined but unknown (for example, the password of an account is encrypted with a fixed password), the cipher text of a character is only related to the character and its location. The cipher text with different contents can be decoded by calculating S_k. S_k can be obtained by simple XOR calculations (because $C = S_k \oplus P$, it is easy to get $S_k = C \oplus P$, only a piece of plaintext with the same length as the target plaintext needs to be prepared for encryption and XOR operation). This also makes RC4 unsuitable for encrypting text with fixed passwords (for example, encrypting user passwords in application systems). The algorithm is also vulnerable to attacks [25].

4) TEA

TEA is a lightweight encryption method which is widely used in IoT and RFID [28]. The method uses a 32-bit key to encrypt data. As a lightweight encryption method, it can run efficiently in embedded systems. Since the key is short, the method is not as safe as traditional encryption methods such as DES and AES.

5) Overview of current encryption methods

These encryption methods use a complicated process to ensure safety. However, the key length decides the actual safety of any encryption method. Unfortunately, the length of a key is often short. Furthermore, the number of encryption algorithms is limited, so hackers can easily decide the algorithm used by checking the cipher text.

III. Definitions and Method

A. Spread activation trajectory and encryption algorithm

The spreading activation method takes a piece of plaintext t (usually a character) as the activation source and generates an activation trajectory as the cipher text by performing two operations called spread and activation in the spreading activation space S. The spread function is used to search the points satisfying the constraint conditions in space S. Once a point satisfies the constraint conditions, the spread operation will be ended immediately, and the relative position and deviation of the point will be recorded as corresponding

cipher text. Then, the activation process will be started to locate the starting search point of the next plaintext.

Definition 1. An n-dimensional spreading activation space is an n-dimensional vector space S. A position in the space can be represented as an n-dimensional vector P_n, which can be abbreviated as P. Each position corresponds to a point in the space, which has a value V_P.

The operation of spread takes the plaintext T and the starting position P as input and finds the first point in the neighborhood of P that conforms to formula 2.

$$p \in \{p \mid p \in S \wedge \left| v_p - t \right| \le \delta\} \tag{2}$$

The δ in formula 2 is the allowed deviation. It can gradually increase with the search process. When the point is found, a tuple c_t described in formula 3 is returned as the corresponding cipher text.

$$c_t = (\Delta p = p - p_0, dev = v_p - t) \tag{3}$$

After finding the qualified point to generate the cipher text, the activation operation is started immediately to determine the starting search position of the next character. The activation operation uses the tuple generated by the spread operation as the input and produces a position p as the next start point. Run the whole process iteratively until all cipher text is generated for the plaintext. The whole encryption algorithm can be described as follows.

Algorithm 1: Encryption Algorithm

Input: plaintext PT, cipher space S, startpoint p
Output: cipher trajectory T
foreach *character c in PT* **do**
  c_t=spread(S,p,c)
  Append c_t to T
  p=activation(c_t)
end
Return(T)

Figure 2 shows the encryption process of the plaintext string "SA" in one-dimensional space. The process is carried out on a one-dimensional integer array with 10 integers and the starting point is the first element (position 0). The spread operation is defined as a forward search until the difference between the ASCII code and the target character is less than or equal to 2. The activation operation is defined as moving forward 3 positions.

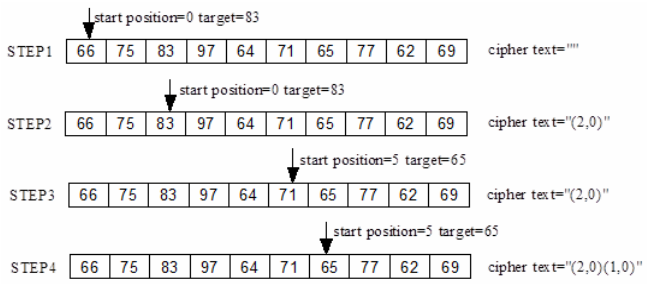

Fig. 2. Spreading activation encryption in 1-D cipher space.

As shown in the figure, the pointer is in position 0 when the target character 'S' comes. The ASCII code of 'S' is 83. Then the pointer shall go to position 2 and the deviation is 0. Tuple (2,0) is the cipher text. In the next step, the pointer moves to position 5 according to the activation function. Target of the second round is character 'A' which has an ASCII code 65. The pointer shall go to position 6. The distance to the start position is 1. So, the cipher text for character 'A' is tuple (1,0).

B. *Decryption algorithm*

It is obvious that the plaintext can be generated by repeating the trajectory in the encryption spreading activation space S. The decryption algorithm is described as follows.

Algorithm 2: Decryption Algorithm

Input: cipher trajectory T, cipher space S, startpoint p
Output: plaintext PT
foreach *tuple* c_t *in P* **do**
  pt=p+ ct.Δp
  c=$V_p t$-c_t.dev
  Append c to PT
  p=activation(c_t)
end
Return(PT)

C. *Efficiency of algorithms*

Although the spreading activation space is large, the encryption algorithms are not so slow in fact. Assuming that there are L kinds of characters uniformly distributed in the spreading activation space, the probability of finding a qualified point in each search can be given by the following formula:

$$\mathrm{p(x)} = \frac{2\delta + 1}{L} \tag{4}$$

The mathematical expectation of times to find the point that meets the requirements is as follows:

$$E(X) = \sum_{i=1}^{\infty} i \times \left[(1 - p(x))^{i-1} - (1 - p(x))^{i} \right] \tag{5}$$

It is easy to prove formula 6:

$$E(X) = \frac{1}{p(x)} \tag{6}$$

When δ = 5, the expectation converges to 23.27. When δ = 10, the expectation converges to 12.19. An average of ten to dozens of searches can complete the encoding of a character. The complexity of the decryption algorithm is O (n). It is efficient and fast.

IV. Cryptanalysis

A. *Enumerating attack*

In the case of knowing the activation function and spreading method, the brute force attacking of spreading activation encryption algorithm needs to guess the encryption space of the algorithm first, then try to verify each point as a potential entrance. The encryption space may be a picture, a file, or even an unbounded space defined by a function. The cost of enumerating is large or even theoretically impossible. If the size of the spreading secret space is limited and denoted as | S |, each point has x values, then the algorithm strength N can be given by the following formula:

$$\mathrm{N} = \mid S \mid \times x^{|S|} \tag{7}$$

Usually, value of x is 256 corresponding to the number of ASCII codes. Clearly, N is a large number when | S | is big.

B. *Choose plain text attack*

Because the cipher text of each character is related to all previous characters, there are 256^i values need to be considered for the attack of the i-th character. For the plaintext with the length of n characters, the encryption program needs to be run 256^n times to enumerate the result. When n = 4, it needs to try 4 billion times to reach the result, which is almost impossible to complete. However, it is almost meaningless to decrypt only the first 3-4 characters.

V. Experiments and Evaluation

A. *Experiments description*

In order to test the performance and security of the algorithm, experiments are carried out in a two-dimensional spreading activation space with appropriate spread operation and deviation parameters. The operation of spread is shown in Figure 3. The starting point is labeled 0 in the center of the space. The surrounding points labeled 1, 2, 3, and so on are searched subsequently until the point meeting the deviation requirements is found.

3	3	3	3	3	3	3
3	2	2	2	2	2	3
3	2	1	1	1	2	3
3	2	1	0	1	2	3
3	2	1	1	1	2	3
3	2	2	2	2	2	3
3	3	3	3	3	3	3

Fig. 3. Spreading order in 2-D cipher space.

Different activation operations and deviation parameters are chosen to generate different algorithms in experiments. The nominations of the algorithms are shown in Table 1.

TABLE I
SPREADING ACTIVATION ALGORITHMS IN EXPERIMENTS

Algorithm	Deviation	Activation function
SA-r-10pd	r	x=px×10×dev;y= py×10×dev
SA-0-10pd	0	x=px×10×dev;y= py×10×dev
SA-1-10pd	1	x=px×10×dev;y= py×10×dev
SA-3-10pd	3	x=px×10×dev;y= py×10×dev
SA-5-10pd	5	x=px×10×dev;y= py×10×dev
SA-10-10pd	10	x=px×10×dev;y= py×10×dev
SA-r-0	r	X=p
SA-r-+20	r	x=x0+20,y=y0+20
SA-r-p	r	x=x0+ px;y=y0+ py

Nine algorithms are used in the experiments. Deviation r is the deviation that equals to the searching layer number of the spread operation (1, 2, 3,…in Figure 2).

A two-dimensional matrix with 1024*1024 elements is used as the spreading activation space in the experiments in this paper. Plaintext used in the experiments is a text file from the web which is 1.09MB. Some experiments use part of the text file. The distribution of characters of the file is shown in Figure 4.

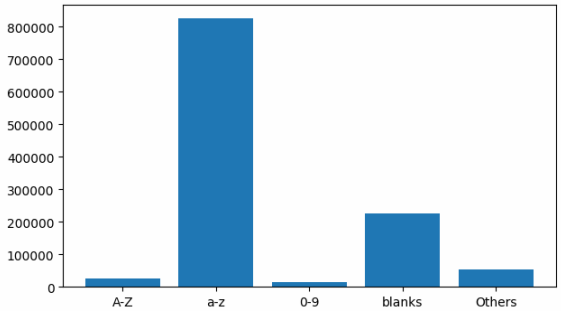

Fig. 4. Distribution of characters in experimental data

B. *Choose activation function*

The activation function has a great influence on the spreading activation trajectory. The experimental results of different algorithms to encrypt a piece of plaintext with about 2000 characters are shown in Figure 5.

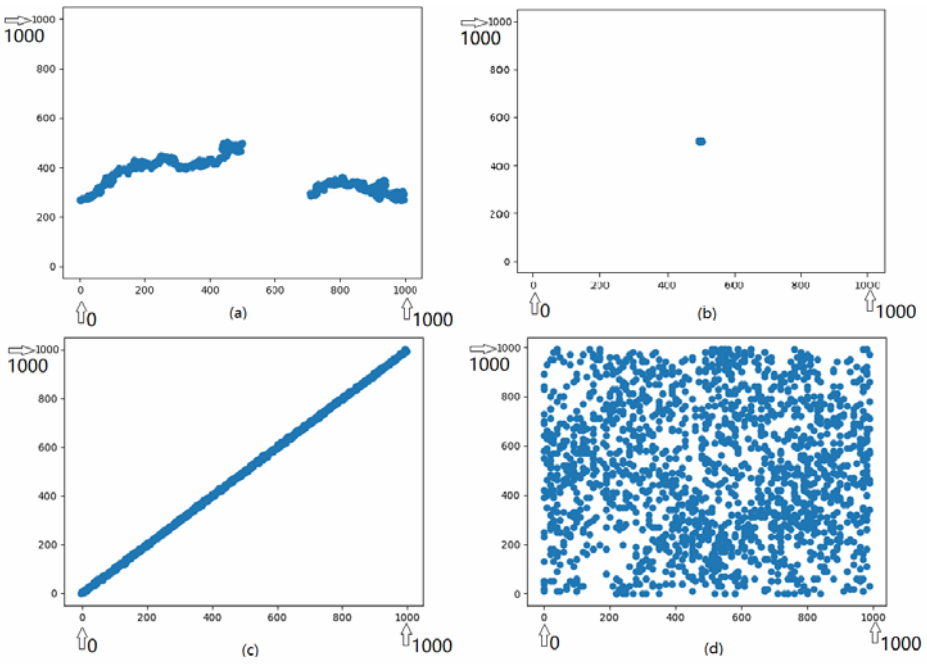

Fig. 5. Activation points in 2-D cipher space

Graph (a), (b) ,(c), (d) in the figure show the distribution of points used for encryption corresponding to algorithm sa-r-p, sa-r-0, sa-r - + 20 and sa-r-10pd respectively. The encryption points of sa-r-0(graph b) distribute in a small range around the starting point. The large encryption space is not well utilized for this algorithm. It is obvious the algorithm sa-r-0 is relatively vulnerable to brute force attack. Algorithm sa-r-p and sa-r-+20 make better use of the encryption space. However, they are not good enough. Especially, the encryption points of sa-r -+20 distribute around a line. It is not a very safe distribution. Algorithm sa-r-10pd's encryption points distribute all over the encryption space evenly. It is difficult to predict and simulate the encryption space used for the activation method.

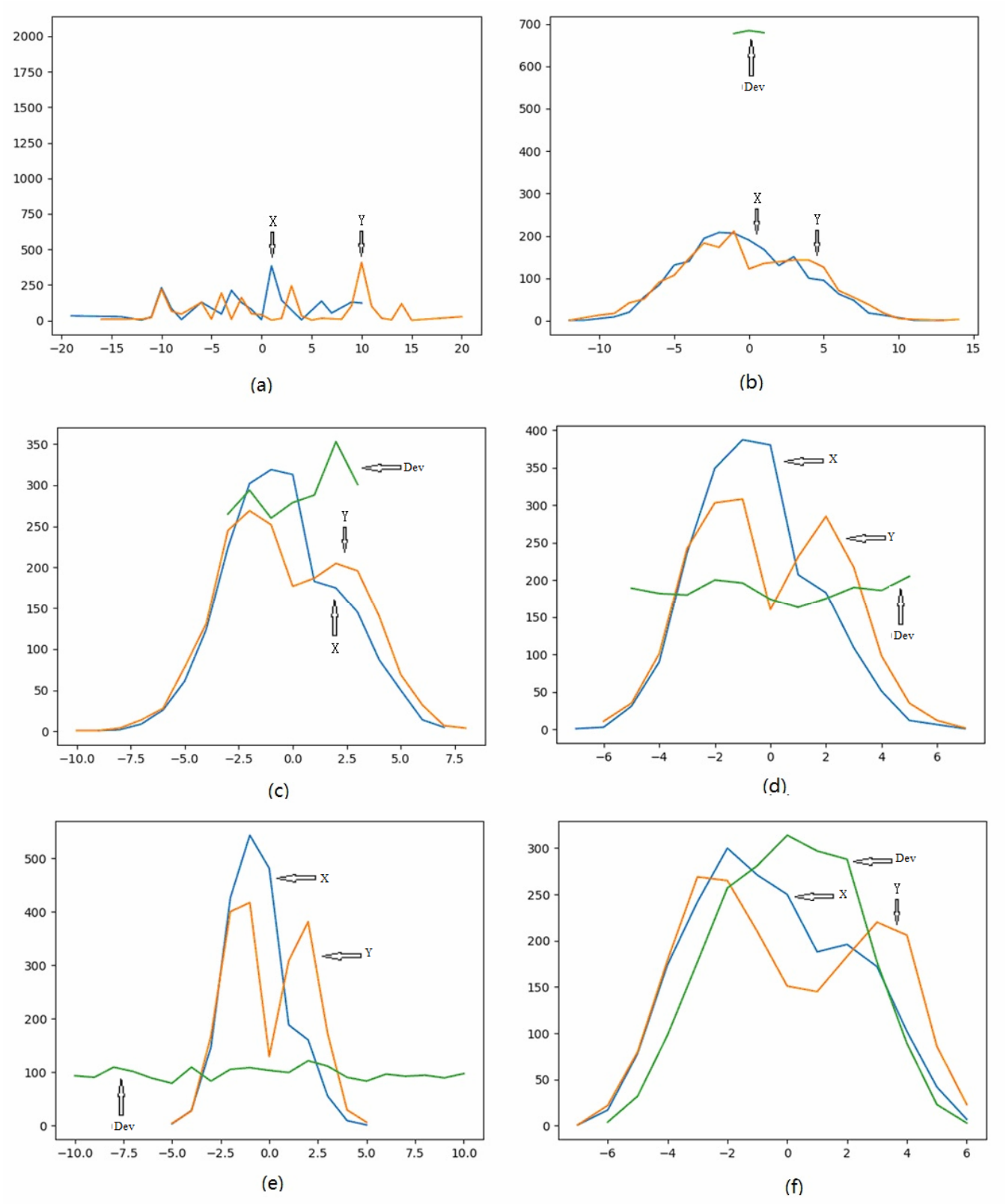

Fig. 6. Distribution of X, Y, Dev in cipher text of different δ

C. Choose deviation parameter

In the encryption process, the selection of the deviation parameter has a great influence on the encryption performance, coding length, and security of the encryption system. In Figure 6, graph (a), (b). (c). (d). (e). (f) are the distribution of output components X, Y, and Dev when the deviation is allowed to be less or equal to 0, 1, 3, 5, 10 and r respectively. Spreading function is the same: $x=p_x\times10\times dev$; $y= p_y\times10\times dev$.

It can be seen from the figure that when the lower deviation parameter is set, the scope of the search becomes larger, and the range of variables X and Y become larger. That means the encryption time will be longer, and X, Y will carry more information. On the contrary, the higher the deviation parameter, the smaller the scope of search, the smaller the value range of X and Y, the shorter the encryption time, and the more information the Dev component carries. The algorithm sa-r-10pd is relatively balanced in all aspects. Furthermore, the distribution also affects the representation of the cipher text. When δ = 0, the value of X and Y varies from – 20 to 20, which requires 6 bits to represent. When δ = 5, they need only 4 bits to represent. Considering conditions such as security, speed and cipher text length, sa-r-10pd, sa-5-10pd and sa-10-10pd are more suitable for practical application.

D. Statistical features of algorithms

Mutual-information and correlation coefficient of plain text and corresponding cipher text is shown in the following table. To get more convincing results, 50 different spreading activation spaces are generated to get a mean score and standard deviation. Because the correlation coefficients can be negative values, mean absolute values are used instead of mean values.

TABLE II
MUTUAL-INFORMATION AND CORRELATION COEFFICIENT OF DIFFERENT ALGORITHMS

Algorithm	Mean Mutual information	Standard deviation	Mean absolute value of correlation coefficients		
			x	y	dev
SA-r-10pd	0.02297	0.00477	0.00455	0.00240	0.00300
SA-1-10pd	0.02488	0.00317	0.00405	0.00530	0.00438
SA-5-10pd	0.02476	0.00368	0.00014	0.00268	0.00248
SA-10-10pd	0.02425	0.00529	0.00314	0.00305	0.00409
SA-r-0	1	0	0.60432	0.28436	0.20456

From the result, it is concluded that the activation operation can solve the problem of statistical vulnerability. With a random cipher space, the first four algorithms in the table are not vulnerable to statistical attacks. Activation operation is not used in the algorithm SA-r-0, so it is vulnerable to statistical attacks.

E. Comparing speed and safety with other methods

Experiments were carried out to compare metrics such as brute force cracking difficulty, encryption speed and decryption speed with other encryption methods. The results are shown in Table 3. The results of spreading activation algorithms are the mean results of 50 experiments.

TABLE III
COMPARE OF DIFFERENT ENCRYPTION ALGORITHMS

Algorithm	Encryption strength	Encryption time (in seconds)	Decryption time (in seconds)
SA-r-10pd	$1024\times1024\times256^{1024*1024}$	10.08	1.41
SA-5-10pd	$1024\times1024\times256^{1024*1024}$	8.02	1.43
SA-10-10pd	$1024\times1024\times256^{1024*1024}$	5.61	1.41
Base64	256^3	1.31	1.35
RC4	256^{12}	1.04	1.25
DES	2^{56}	54.17	52.12
AES-CBC	2^{192}	8.67	6.49
TEA	2^{32}	8.32	8.31

In order to compare the efficiency of each algorithm fairly, the data in the table are obtained by programs in python using the same plaintext which has the size of 1.09MB. The experimental results show that the encryption strength of the spreading activation encryption algorithm is high. The encryption speed is relatively slow but faster than DES. The encryption and decryption speed of TEA are both slower than those of the spreading activation based method.

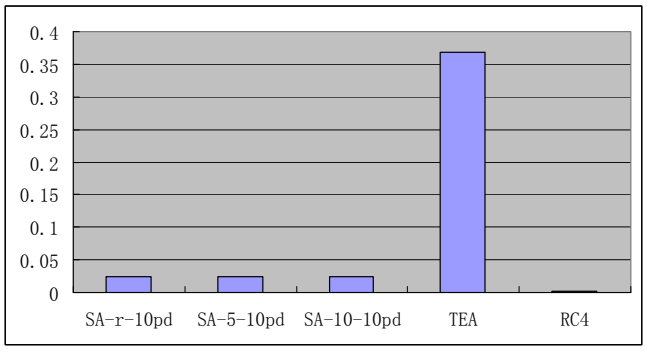

Fig. 7. Mutual-information scores of lightweight encryption algorithms.

Figure 7 shows the mutual-information scores of lightweight encryption algorithms. The lower score means the less relation between plaintext and cipher text. All the spreading activation based algorithms get similar mutual information scores. These scores are much lower than the score of TEA. Although they are a little higher than that of RC4 algorithm, they are low enough to resist the attack based on statistics.

VI. CONCLUSION

Many algorithms can be derived from the spreading activation encryption method. These algorithms can use large spreading activation encryption space and more complex operation of spread and activation to encrypt. The value of a packet affects the cipher text of all subsequent packets in all these algorithms. The time cost of brutal cracking these algorithms can be great. The mutual information scores between plaintext and cipher text in many algorithms are low enough to resist the statistical cryptanalysis of known plaintext and cipher text. In addition, they cannot be described by simple logic functions because the activation and spreading operations are variable. It is difficult to find a

common weakness of spreading activation based algorithms.

On the other hand, although the keys can be very long, spreading activation based algorithms can be run efficiently especially in the decryption process.

These algorithms are suitable both for text file and binary file encryption.

VII. DISCUSSION AND FUTURE WORKS

A. Selection of encryption space

The construction and selection of encryption space are also important factors affecting the distribution of cipher text and security. Taking this experiment as an example, it is obvious that the proportion of blanks and lowercase letters is much larger than that of uppercase letters. If the distribution of values in the encryption space is similar to the distribution of ASCII codes in plaintext, the speed of the encryption algorithm will be much faster. However, the security decreases while fewer values are contained in the encryption space. The method will be more vulnerable to side-channel attacks [27] too.

There are also some bad encryption spaces just as weak keys in other encryption methods. Obviously, an encryption space consisting of points with the same value is a bad one. Encryption spaces consists of points with similar values in their neighborhoods are not good too. However, the activation process can solve the problem. For example, an experiment applying sa-r-10pd on the encryption space generated according to the picture *Lenna* results in the mutual information 0.07246. The result is worse than those results with random-valued encryption spaces but acceptable. According to the experiments, randomly generated encryption spaces are often good enough.

B. Good or best algorithms

The algorithms in this paper such as sa-r-10pd, sa-5-10pd and sa-10-10pd are effective and efficient, but they may not be the best algorithms in the spreading activation scheme.

Many spreading activation based algorithms are good encryption algorithms. Although it is not easy to find the unique best combination of spreading space and activation operation, these algorithms are good enough as encryption algorithms.

REFERENCES

[1] J. R. Anderson, P. L. Pirolli, "Spread of Activation," *Journal of Experimental Psychology: Learning Memory and Cognition*, vol. 10, no. 4, pp. 791-798, 1984.

[2] J. Balaji, T. V. Geetha, P. Ranjani, "Abstractive Summarization: A Hybrid Approach for the Compression of Semantic Graphs," *International Journal on Semantic Web and Information Systems*, vol. 12, no. 2, pp. 76-99, 2016.

[3] A. M. Collins, F. E. Loftus, "A Spreading-Activation Theory of Semantic Processing," *Psychological Review*, vol. 82, no. 6 , pp. 407-428, 1975.

[4] M. de Gemmis, P. Lops, G. Semeraro, C. Meto, "An Investigation on the Serendipity Problem in Recommender Systems," *Information Processing & Management*, vol. 51, no. 5, pp. 695-717, 2015.

[5] G. S. Dell, "A Spreading-Activation Theory of Retrieval in Sentence Production," *Psychological Review*, vol. 93, no. 3, pp. 283-321, 1986.

[6] X. Fu, H. Wei, "Problem Solving by Soaking the Concept Network," *Computer Science and Information Systems*, vol. 8, no. 3, pp. 761-778, 2011.

[7] X. H. Fu, W. W. Liu, Y. Y. Xu, L. Z. Cui, "Combine Hownet Lexicon to Train Phrase Recursive Autoencoder for Sentence-Level Sentiment Analysis," *Neurocomputing*, vol. 241, pp. 18-27, 2017.

[8] A. Fuster-Sabater, P. Garcia-Mochales, "On the Balancedness of Nonlinear Generators of Binary Sequences," *Information Processing Letters*, vol. 85, no. 2, pp. 111-116, 2003.

[9] G. Giray, M. O. Unalir, "Assessment of Text Coherence Using an Ontology-Based Relatedness Measurement Method," *Expert Systems*, vol. 37, no. 3, pp. 24, 2020.

[10] G. Goth, "DES Is Dead: Nist Declares Standard Officially Obsolete," *IEEE Security & Privacy*, vol. 2, no. 5, pp. 9, 2003.

[11] M. E. Hameed, M. M. Ibrahim, M. N. Abd, A. A. Mohammed, "A Lossless Compression and Encryption Mechanism for Remote Monitoring of ECG Data Using Huffman Coding and CBC-AES," *Future Generation Computer Systems*, vol. 111, pp. 829-840, 2020.

[12] J. Kim, H. Seokhie, H. Dong-Guk, L. Sangjin, "Improved Side-Channel Attack on DES with the First Four Rounds Masked," *ETRI Journal*, vol. 31, no. 5, pp. 625-627, 2009.

[13] A. Klein, "Attacks on the RC4 Stream Cipher," *Designs Codes and Cryptography*, vol. 48, no. 3, pp. 269-286, 2008.

[14] J. H. Lee, D. G. Han, "Security Analysis on Dummy Based Side-Channel Countermeasures-Case Study: AES with Dummy and Shuffling," *Applied Soft Computing*, vol. 93, pp. 106352, 2020.

[15] D. B. Lenat, "CYC - a Large-Scale Investment in Knowledge Infrastructure," *Communications of the ACM*, vol. 38, no. 11, pp. 33-38, 1995.

[16] M. Masoumi, "Novel Hybrid Cmos/Memristor Implementation of the AES Algorithm Robust against Differential Power Analysis Attack," *IEEE Transactions on Circuits and Systems*, vol. 67, no. 7, pp. 1314-1318, 2020.

[17] J. D. Ming, Y. B. Zhou, W. Cheng, "Mind the Balance: Revealing the Vulnerabilities in Low Entropy Masking Schemes," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3694-3708, 2020.

[18] T. M. Mitchell, "Generalization as Search," *Artificial Intelligence*, vol. 18, no. 2, pp. 203-226, 1982.

[19] W. Mula, D. Lemire, "Faster Base64 Encoding and Decoding Using Avx2 Instructions," *ACM Transactions on the Web*, vol. 12, no. 3, pp. 1-26, 2018.

[20] A. Reyhani-Masoleh, M. Taha, D. Ashmawy, "New Low-Area Designs for the AES Forward, Inverse and Combined S-Boxes," *IEEE Transactions on Computers*, vol. 69, no. 12, pp. 1757-1773, 2020.

[21] R.Saha, G. Geetha, G. Kumar, "MRC4: A Modified RC4 Algorithm Using Symmetric Random Function Generator for Improved Cryptographic Features," *IEEE Access*, vol. 7, pp. 172045-172054, 2019.

[22] C. E. Shannon, "Communication Theory of Secrecy Systems," *Computing*, vol. 15, no. 1, pp. 57-64, 1998.

[23] C. S. Q. Siew, "Spreadr: An R Package to Simulate Spreading Activation in a Network," *Behavior Research Methods*, vol. 51, no. 2, pp. 910-929, 2019.

[24] M. Sigman, G. A. Cecchi, "Global Organization of the Wordnet Lexicon," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 3, pp. 1742-1747, 2002.

[25] Y. Tsunoo, T. Saito, H. Kubo, "A Distinguishing Attack on a Fast Software-Implemented RC4-Like Stream Cipher," *IEEE Transactions on Information Theory*, vol. 53, no. 9, pp. 3250-3255, 2007.

[26] T. Vileiniskis, R. Butkiene, "Applying Semantic Role Labeling and Spreading Activation Techniques for Semantic Information Retrieval," *Information Technology and Control*, vol. 49, no. 2, pp. 275-288, 2020.

[27] A. Levina, V. Varyukhin, D. Kaplun rt al., "A Case Study Exploring Side-Channel Attacks On Pet Wearables," *IAENG International Journal of Computer Science*, vol. 48, no. 4, pp. 878-883, 2021.

[28] Mishra, Z. , B. Acharya . "High throughput novel architectures of TEA family for high speed IoT and RFID applications," Journal of Information Security and Applications vol. 61, no. 7, pp. 102906, 2021.

Xixu Fu (M'20) got his Ph.D of computer software and theory in Fudan University at 2015. Now, he is a lecturer of Shanghai Ocean University. He became a Member (M) of IAENG in 2020. His research interests include artificial intelligence, security and data science.

Hu Li got his Ph.D of computer software and theory in Fudan University at 2013. Now, he is a lecturer of Shanghai Lixin University of Accounting and Finance. His research interests include artificial intelligence, computer vision and cryptography.