HCCA: A Cryptogram Analysis Algorithm Based on Hill Climbing

Zhang Tongbo
College of Computer Science and Technology
Jilin University
Changchun, China
ztb5129@live.com

Xu Yue
College of Computer Science and Technology
Jilin University
Changchun, China
phoenix_sands@live.com

Li Guangli
College of Computer Science and Technology
Jilin University
Changchun, China
calculatinggod@foxmail.com

Weng Jie
College of Computer Science and Technology
Jilin University
Changchun, China
jerrywossion@gmail.com

Lu Shuai *
College of Computer Science and Technology
Jilin University
Changchun, China
lus@jlu.edu.cn
* Corresponding Author

Abstract—The single letter substitution encryption is the basis of the most widely encryption methods in cryptography. Therefore, it is extremely significant for the development of the cryptography to decipher this kind of encryption efficiently and correctly. Researchers elaborately analyzed the features of frequency analysis algorithm and the pattern matching algorithm as well as combined the strengths of each algorithm. Aiming at the circumstance that the cryptogram transmission channel has some noise interference, researchers finally designed a cryptanalysis algorithm HCCA based on hill climbing algorithm, according to the statistical regularities of nature language and the pattern characteristic of different words, which is on basis of the two algorithms mentioned above. The results of experiment showed that the cryptanalysis algorithm HCCA could decrypt the substitution cipher efficiently and correctly. In addition, the cryptanalysis algorithm HCCA could still complete the decryption work correctly under the circumstance, and there exists some noise interference in different degrees.

Keywords-Hill Climbing; Pattern Matching; Frequency Analysiss; Substitution Cipher; Cryptogram Analysis

I. INTRODUCTION

The cryptography mainly consists of the cipher-coding and the cryptanalysis. The primary mission of cipher coding is information shielding by coding the information. And the cryptanalysis primarily researches on the plaintext information acquisition by analyzing the cryptogram. These two theories collectively promote the development of the cryptography. Moreover, the research of the cryptanalysis mainly focuses on the strong attack, current cryptanalysis [1], differential cryptanalysis [2-5] and so on. The single letter substitution encryption is the basis of the most part of encryption methods in cryptography. So it is extremely significant for the development of the cryptography to decipher this kind of encryption efficiently. Aiming at the decryption of the single letter substitution encryption, researchers brought up a cryptanalysis algorithm HCCA based on the Hill Climbing and compared the frequency analysis with the pattern matching.

II. FREQUENCY ANALYSIS

In cryptography, the frequency analysis [6-7] researches on the frequency of letters or monograms appeared in the text. It can be found by analyzing a large amount of English literature that the relative frequency of the appearance of letters is stable. The laws worked out by frequency analysis of the modern English are as follows:

- The correspondence rules of single letter (descending order): E, T, A, O, N, R, I, S, H, D, L, F, C, M, U, G, Y, P, W, B, V, K, J, X, Q, Z;
- The correspondence rules of the bigram (descending order): TH HE, AN, IN, ER, ON, RE, ED, ND, HA, AT, EN, ES, OF, NT, EA, TI, TO, IO, LE, IS, OU, AR, AS, DE, RT, VE, ON, ST, NT, NG, OR, ET, IT, AR, TE, SE, HI;
- The correspondence rules of the trigram (descending order): THE, AND, THA, ENT, ION, TIO, FOR, NDE, HAS, NCE, TIS, OFT, MEN;
- The correspondence rules of the quad grim (descending order): THAT, THER, WITH, DTHE, NTHE, OTHE, OFTH, TTHE, FTHE, TION, THES, EAND, HERE, INGT, ANDT, SAND, ETHE, THEM, THEC, NDTH, TOTH;

TABLE 1 SINGLE LETTER FREQUENCY

Letter	Frequency	Letter	Frequency
A	8.167	**O**	7.507
B	1.492	**P**	1.929
C	2.782	**Q**	0.095
D	4.253	**R**	5.987
E	12.702	**S**	6.327
F	2.228	**T**	9.056
G	2.015	**U**	2.758
H	6.049	**V**	0.978
I	6.966	**W**	2.360
J	0.153	**X**	0.150
K	0.772	**Y**	1.974
L	4.025	**Z**	0.074
M	2.406		

In the single letter substitution encryption, every letter is substituted by another letter, and the same letter in plaintext is always substituted the corresponding letter. The certain statistical characteristics existed in plaintext are still reserved in cryptogram. By the statistics of the frequency distribution of the letters or monograms in cryptogram and the research on the relationship between the letters, researchers then contrasted with the frequency distribution of the letters or monograms in modern English, corresponded between the high frequency letter monograms in cryptogram and that in laws. So that researchers can build the one-to-one mapping between the letter in cryptogram and the letter in laws and decipher the substitution cipher in the end.

Consider that the intervals and punctuations in plaintext are all deleted after the encryption, it's necessary to segment the cryptogram by different lengths. In order to obtain all the continuous letter monograms in cryptogram, researchers segmented the cryptogram by dislocation segmentation.

EXAMPLE 1. The cryptogram has a length of m, researchers want to obtain all the continuous letter monograms which length is n after segmentation. The segmentation processes in n times:

The first time: Segment and save the cryptogram in every n characters from the head.

The second time: Segment and save the cryptogram in every n characters from the second character.

The last time: Segment and save the cryptogram in every n characters from the nth character.

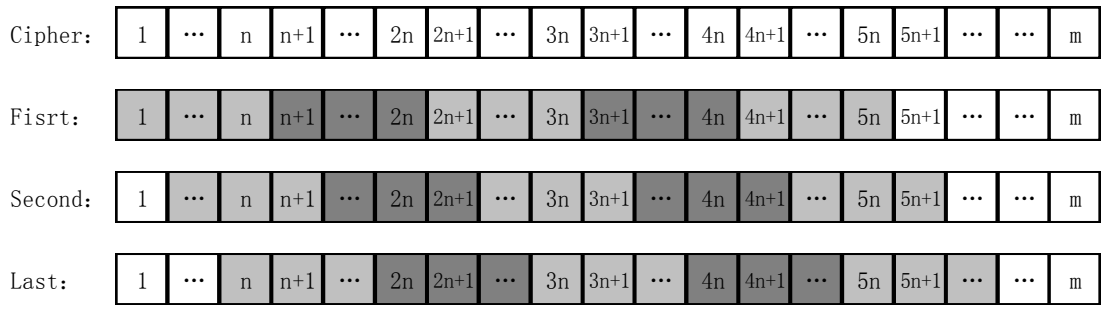

Figure 1. Dislocation segmentation

It is known by the characteristics of the single letter substitution encryption that KEY is a string of 26 bits, and the i^{th} bit represents changing the i^{th} letter in standard alphabet of plaintext into the i^{th} letter of the KEY. In order to get the KEY, researchers built up a waiting queue sorted by probability to every bit of the KEY, and stored these queues into a 26*26 waiting matrix W_Freq. The value of W_Freq[i][j] represents probability that the i^{th} bit of the KEY is letter j. According to the laws set up above, going through the frequency analysis by cryptogram, counting up the probability of certain monograms appeared in cryptogram and sorts them in descending order. Compare the sorted results with the laws above, supposed that the monograms in same sequential position are the corresponding relations of the cryptanalysis, and update waiting matrix in turn.

The algorithm flow chart is as follows:

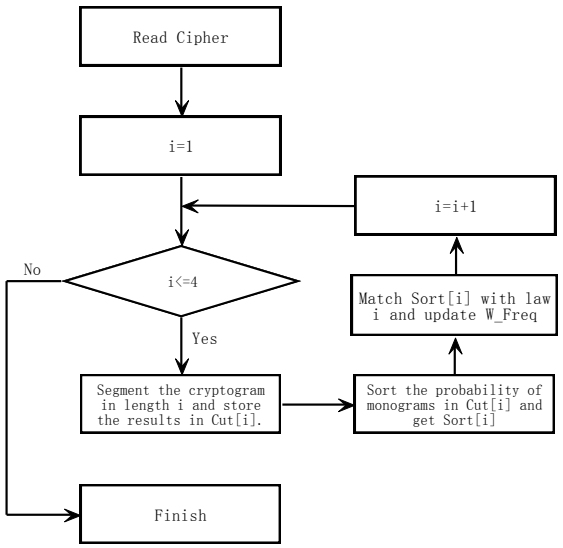

Figure 2. Flow chart of frequency analysis

III. PATTERN MATCHING

The foundation of pattern matching is the limited English vocabulary, and the letters in words are all sorted in some rules, not randomly.

EXAMPLE 2. "Attract" and "osseous" are the only two "1223142" patterns in English. That is to say, if there is a "1223142" pattern in cryptogram, researchers can guess it as "attract" or "osseous". According to this principle, screen the word patterns in cryptogram in a special word pattern library, and then researchers will find the most matching KEY.

In order to build the word pattern base researchers need, it's necessary to subtotal the common vocabulary by length and the letter sequence. Then generate the pattern matching library. After modeling there are some patterns. At the same time, count up the corresponding word frequency of every pattern. Then researchers will get a "pattern - frequency - pattern" list as the word pattern base. Consider the number of patterns which length is less than 2 is so large that makes no difference to the actual matching. Therefore, researchers delete it in the word pattern base. Consider the number of pattern that the length is less than 2 and it is so large that make no difference with the actual matching. So researchers delete it in the word pattern base. The final word pattern base has 1935 kinds of patterns. Parts of the patterns are as follows:

TABLE 2 WORD PATTERN BASE

Pattern	Frequency	Pattern	Frequency
111	6	**1122**	2
112	46	**1123**	37
121	94	**1211**	19
122	72	**1212**	51
123	1749	**1213**	235
1111	2	**1221**	26
1121	2	**1222**	5
1223	310	**1233**	340
1231	294	**1234**	5586
1232	357	**11213**	6

In order to get the KEY, built up a waiting queue sorted by probability to every bit of the KEY, and stored these queues into a 26*26 waiting matrix W_Freq. The value of W_Freq[i][j] represents probability that the i^{th} bit of the KEY is letter j. According to the word pattern base above, going through the pattern matching by cryptogram, if has matched the fit pattern, for example, if researchers got the "1223142" pattern in the cryptogram, researchers can get it from the word pattern base that there are only two words in this pattern which are "attract" and "osseous". It showed that the letter in position "1" has a probability of 1/2 to be "a", and the other 1/2 is "o", then update the W_Freq waiting queue by this.

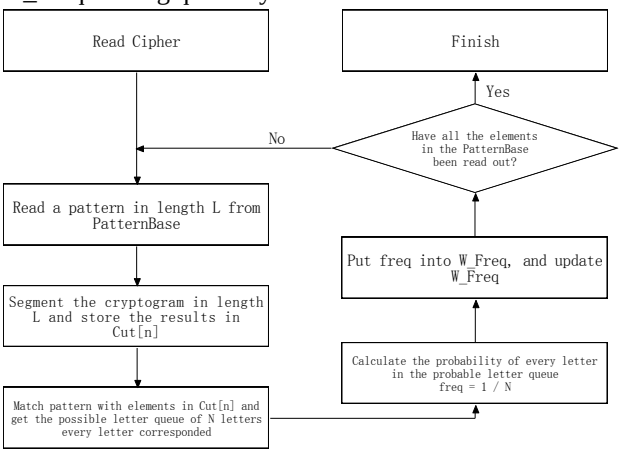

Figure 3. Flow chart of pattern matching

IV. HCCA – THE CRYPTOGRAM ANALYSIS ALGORITHM BASED ON HILL CLIMBING

We could see that the two kinds of decipher methods above have some limitation. So researchers consider combining their superiorities and designing a comprehensive cryptanalysis algorithm. This algorithm skillfully brings in the hill climbing algorithm while deciphering. Hill climbing [10-11] each time chooses an optimal solution as current solution from the near-optimal solution space until achieving a local optimal solution. The main shortcoming is that researchers may get into the local optimal solution but not always achieve the global optimal solution. It is shown in the following chart: Suppose Point C as the current solution, the hill climbing algorithm will stop search when gets the local optimal solution (Point A). Since Point A can't get a better solution whichever direction it moves.

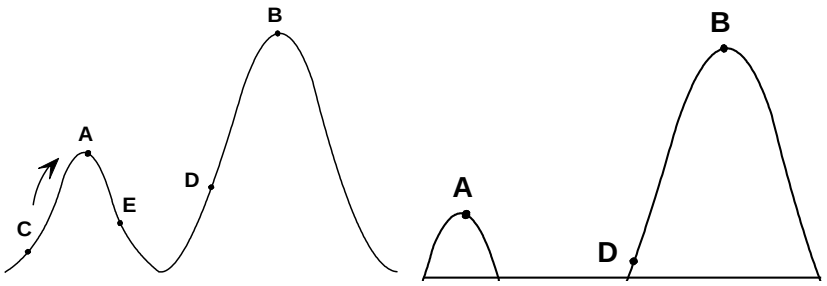

Figure 4. Hill climbing algorithm

The comprehensive cryptanalysis algorithm made the result that generated from the two decipher methods as the beginning of the hill climbing algorithm, which decreased the height gap between the top and the bottom. So that a small move could make it from point A to point D, which reduced the possibility that Hill Climbing fell in the local optimal solution because of the self-defeat.

There are the pseudo-codes of the algorithm:

TABLE 3 ALGORITHM 1. HCCA

```
HCCA-CIPHER-SOLVER (Cipher)
1   Load Rules
2   Load PatternBase
3   W_freq_rate = FREQUENCY-ANALYSIS-MODEL(Cipher)
4   W_freq_pat = PATTERN-MATCHING-MODEL(Cipher)
5   W_freq = COMBINE(W_freq_rate, W_freq_pat)
6   BestKey = HILL-CLIMBING(Cipher, W_Freq)
7   return BestKey
```

```
HILL-CLIMBING(CIPHER, KEY_ARRAY)
1   while( i < AttemptTimes )
2       for (int y = 0; y < (i * 2) - 1; y++)
3           KEY_Array[i].Shuffle()
4           (KEY_Array[i],Score) = FindKey(KEY_Array[i], Score, Cipher)
5           if (score < bestScore)
6                   bestKey =KEY_Array[i]
7                   bestScore = score
8                   if(bestScore < SCORE_min)
9                           return bestKey
10  return bestKey
```

In Algorithm SAHC-CIPHER-SOLVER, researchers first generated W_Freq(lines 1-5) based on Model 1 and Model 2. Then researchers got BestKey according to Function HILL-CLIMBING.

In Algorithm HILL-CLIMBING, researchers searched every KEY in the KEY_Array (lines 1-15). For simulating the jitter of a certain probability in algorithm, researchers built Function Shuffle (line 4). Function Shuffle swaps two positions of KEY every time to simulate the small move in algorithm. Store the evaluation of KEY every time into SCORE, and estimate the probability that can find the translated plaintext in alphabet (line 5). Function return BestKey in the end (line 16).

V. PERFORMANCE EVALUATION

In this part, researchers encrypted part of the text from the John Kennedy's inaugural speech and generated the cryptogram and short cryptogram as the test data. The experiment circumstance: hardware CPU: i7-3770, RAM: 8 G; software Windows 8/Visual Studio 2013.

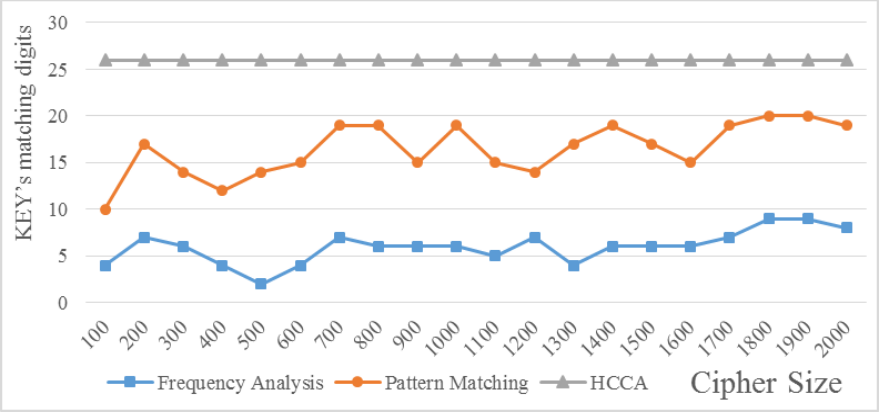

Figure 5. Evaluation of KEY's matching rate

A. *Evaluation of KEY's Matching Digits*

It can be found in analyzing the results above that the KEY matching digits generated from the frequency analysis algorithm floats slightly, but the integral level is low, and the matching digits are all lower than 10. The pattern matching algorithm is better than frequency analysis algorithm on the whole, but there is also slightly float. And these two kinds of algorithm too rely on the quality of cryptogram while deciphering, so they can't decipher the cryptogram effectively when the quality of cryptogram is poor. So here comes the HCCA. It combines the two algorithms above and gets both superiorities. It

works out well no matter the length of the cryptogram is long or short.

B. Evaluation of the Influence that Interference Made to KEY Matching Rate

In this experiment, p1 represents the probability that character is missed during the transmission, p2 represents the probability that character is normally transported but is added a random character after, p3 represents the probability that character is tampered with a random character during the transmission, 1-p1-p2-p3 represents the probability that character is transported normally. Consider the actual cryptogram transmission interference extent, set the minimum of the interference extent as 1%, set the maximum as 10%, set the stepping as 1%, on circumstance of long cryptogram and short cryptogram, test the model by the standard of KEY matching digits. The length of the long cryptogram is 2000, the short is 170.

TABLE 4 Evaluation of the influence

Interference extent (%)			**KEY matching (bit)**	
P1	**P2**	**P3**	Long cryptogram	Short cryptogram
1	1	1	26	24
2	2	2	26	22
3	3	3	26	22
4	4	4	26	21
5	5	5	26	18
6	6	6	26	6
7	7	7	26	6
8	8	8	24	4
9	9	9	24	4
10	10	10	21	4

It's shown in the result that when the cryptogram was long, the interference extent made little difference to the KEY matching digits. But when the cryptogram was short, KEY's matching extent more than 5% when the KEY matching digits declined instantly, and at this moment the algorithm made no contribution to the decipher. So the KEY researchers got had no reference value at all.

VI. CONCLUSIONS

On basis of the further research of frequency analysis and pattern matching algorithm, researchers combined the superiorities of the two algorithms, brought up a comprehensive cryptanalysis algorithm based on the hill climbing algorithm. And researchers achieved the decipher program according to the comprehensive cryptanalysis algorithm, tested the normal cryptogram and the cryptogram interfered by noise. The experiment showed that in all circumstances the algorithm could enhance the decipher efficiency sharply so that the single letter substitution encryption could be deciphered effectively.

ACKNOWLEDGMENT

Project supported by the National Nature Science Foundation of China (No. 61300049), the Specialized Research Fund for the Doctoral Program of Higher Education of China (No. 2012006112005), the China Postdoctoral Science Foundation (No. 2011M500612), the Key Program for Science and Technology Development of Jilin Province of China (No. 20130206052GX) and the Natural Science Research Foundation of Jilin Province of China (No. 20140520069JH, No. 20150520058JH).

REFERENCES

[1] Cho J Y, "Linear cryptanalysis of reduced-round PRESENT," Topics in Cryptology-CT-RSA 2010, pp. 302–317.

[2] Yin G L, Wei H R, "Impossible differential cryptanalysis in CLEFIA algorithm," Computer Science, 2014, vol. Z1, pp. 352–356 (in Chinese with English abstract).

[3] Chen J, Zhang Y Y, Hu Y P, "A new impossible differential cryptanalysis method with six-wheeled AES," Journal of Xidian University, 2006, vol. 4, pp. 598–601 (in Chinese with English abstract).

[4] Zhang W T, Wu W L, Zhang L, "Aiming at related-key about low wheel AES-256 - impossible differential cryptanalysis," Journal of Software, 2007, vol. 11, pp. 2893–2901 (in Chinese with English abstract).

[5] Guo J S, Luo W, Zhang L, "Impossible differential cryptanalysis in LBlock code," Journal of Electronics and Information, 2013, vol. 6, pp. 1516–1519 (in Chinese with English abstract).

[6] Shrivastava G, Sharma R, Chouhan M, "Using Letters Frequency Analysis in Caesar Cipher with Double Columnar Transposition Technique," International Journal of Engineering Sciences and Research Technology, 2013, vol. 2, pp. 1475-1478.

[7] Ziatdinov M. Using frequency analysis and Grover's algorithm to implement known ciphertext attack on symmetric ciphers[J]. Lobachevskii Journal of Mathematics, 2013, vol. 34, pp. 313-315.

[8] Mishra S, Bhattacharjya A, "Pattern analysis of cipher text: A combined approach," Recent Trends in Information Technology (ICRTIT), 2013 International Conference on. IEEE, 2013, pp. 393–398.

[9] Tomohiro I, Inenaga S, Takeda M, "Palindrome pattern matching," Combinatorial Pattern Matching. Springer Berlin Heidelberg, 2011, pp. 232–245.

[10] Zhang X L, Li Q, Yin M H, "A improved hill climbing algorithm with stop mechanism," Proceedings of the CSEE, 2012, vol. 14, pp. 128–134 (in Chinese with English abstract).

[11] Liu Y, Ma J, Chen J, "Robustness properties of hill-climbing algorithm based on Zernike modes for laser beam correction," Applied optics, 2014, vol. 53, pp. 140-146.