An Improved IDEA-ECB Based on Multithread Technology

Gen-Qing BIAN\textsuperscript{a}, Yu-Yao WANG\textsuperscript{b,*}

School of Information and Control Engineering, Xi’an University of Architecture and Technology, Xi’an, Shannxi, China
\textsuperscript{a}bgq_00@163.com, \textsuperscript{b}1924867180@qq.com

*Corresponding author

Keywords: Data Encryption, DEA, Electronic Code Book (ECB), Multithreading.

Abstract. In terms of the problem that the encryption rate of International Data Encryption Algorithm (IDEA) is low, and it’s throughout capacity is small under the ECB packet mode, this paper proposes a kind of improved IDEA-ECB based on multithread technology. Change the order flow of single thread into concurrent flow of multithread with clocking plaintext and multithread technology, which improve the encryption rate and throughout capacity, at the same time reduce the operation hours of encryption algorithm, and satisfy the demand that real time and high effectively software encryption in instant messaging. Experimental results show that if the size of plaintext is 1MB, the number of threads which is started respectively is 4, 8, 12, 16, the speedup of improved algorithm increases 1.72~4.08 times than classical algorithm.

Introduction

International Data Encryption Algorithm [1, 2] (IDEA) was first proposed by Chinese young scholars Lai X who lives in Switzerland and James Massey who is a famous cryptography expert in 1990. In 1992, it was improved to strengthen the ability of anti-differential analysis and renamed International Data Encryption Algorithm. IDEA algorithm is one of the most widely used block encryption algorithms. So far, there are no successful attack reports about it. So now it seems that the IDEA is very safe.

Since the IDEA encryption algorithm was put forward, the current study of the IDEA focuses on three aspects: security, hardware implementation and implementation efficiency of algorithm software. According to the key characteristic of IDEA, references [3] used the relevant key and differential linear to attack the IDEA. Reference [4] improved a sub key expansion algorithm of IDEA, and the algorithm used the pseudo-random sequence to generate the disorder of the key, destroyed attack condition of targeted attacks and obtained more efficient safety. References [5] put forward respectively a kind of IDEA to realize hardware applications, obtaining high resource utilization and throughput. So far, in the field that satisfying the demand that real time and high effectively software encryption in instant messaging, concrete realization methods in term of running high efficiency of software based on IDEA are not exist.

In order to improving the encryption processing performance of IDEA-ECB and satisfying the demand that high speed real time communication needs high speed cryptography algorithm in software field, this paper puts forward an improved IDEA-ECB based on multithread technology.
The ECB Working Mode of IDEA Encryption Algorithm

IDEA Encryption Algorithm

The packet length of IDEA is 64 bits and the key length is 12 bits. IDEA is an iterative algorithm, which is made of 8 rounds that each round has 6 sub key and one output that includes 4 sub key. Each round consists of three functions: modular $2^{16} + 1$ multiplication $\Box$, modular $2^{16}$ additive operation $\oplus$ and bit wise XOR operation $\oplus$. IDEA uses the mixed operation from different algebraic groups [6], the group operation of the three kinds of functions and the structure of multiplication and password to achieve the confusion, so that the complex structure of the IDEA cannot be simplified. Its specific encryption processes are as follows:

1) The key generation. 52 sub keys are generated by 128 bits original key and every sub key contains 16 bits, use $k = G_k(128)$ to generate sub key $k^{(i)}$, $G_k$ is the shift operation generated by sub key.

2) The encryption operation. 64 bits is divided into 4 sub groups: X1, X2, X3, and X4. These 4 sub groups become the first round of the algorithm, a total of 8 rounds. In each round, the encryption process is own in Fig. 1.

3) The encryption cycle. The output of each round forms 4 sub groups (Y1, Y2, Y3 and Y4), between the round and the round, exchange the middle of the two packets and take it as the next round of the input, except the last round. After 8 operations, the final output is obtained.

IDEA-ECB Mode

Code Book Electronic [7] (ECB) mode is the most basic working mode of block cipher. In this mode, the information to be processed is divided into groups, which are appropriate for the size, and then encrypt or decrypt each packet independently, and the work mode is shown in Fig. 2.
The process of IDEA-ECB encryption is shown in algorithm 1.
Algorithm 1: the process of IDEA-ECB encryption
Input: $n$ plaintext group, $P_j \in (P_1, P_2,..P_n)$, $0 \leq j \leq n$;
Output: $n$ ciphertext group, $C_j \in (C_1, C_2,..C_n)$, $0 \leq j \leq n$;
Specific steps of the algorithm:
- After encryption $C_j = E_k(P_j)$, $E_k$ is the encryption operation of the key $K$;
- After decryption $P_j = D_k(C_j)$, $D_k$ is the decryption operation of the key $K$.
IDEA-ECB is that because all operations are controlled by the main thread serial processing, each group has to wait for the completion of the 8 round of operations and then under next processing, for ECB encryption mode, its processing efficiency is low, and it is not suitable for the application of high speed of encryption and decryption.

The Improved Design of IDEA-ECB Based on Multithread Technology

The thread is the independent and concurrent execution flow in program [8]. It can include multiple non-interfering sequential flows. The thread has 5 kinds of state transition in total, and the states respectively are new, runnable, running, blocked, and dead. Accompanied by the running program, a variety of states transform into each other, and it realizes multithreading concurrent execution. Considering decryption process and encryption process of the IDEA is similar [9], this paper involved IDEA takes the encryption process as examples.

The process of improved IDEA-ECB encryption is shown in algorithm 2.
Algorithm 2: the process of improved IDEA-ECB encryption
Input: $n$ plaintext group, $P_j \in (P_1, P_2,..P_n)$, $0 \leq j \leq n$ , key $k$;
Output: $n$ ciphertext group, $C_j \in (C_1, C_2,..C_n)$, $0 \leq j \leq n$;
Specific steps of the algorithm:
1. use random function of Java to generate plaintext randomly, and divide the plaintext into $n$ groups according to the size of 64 bits;
2. use $k = G_k(128)$ to generate the key $k^{(i)}$, as 52 key, and 128 is the key length of IDEA;
3. implement the Runnable interface, set up the thread management parameters $\xi_j$, and rewrite the run () method;
④ Start a thread, perform concurrent encryption operations, \( \xi_j \rightarrow C_j = E_k(P_j) \), \( E_k \) it is a key cryptographic operations, and control a variety of state of threads transformation. The work mode is shown in Fig. 3.

![Diagram](image)

Figure 3. IDEA-ECB Encryption Process on Multithread Technology.

Set up the management parameters \( \xi_j \) of the main thread and specify programs containing different execution threads. Each thread has its own method to call stack and the program counter, making thread and other threads execute concurrently to share resources. Through the organic dispatch and control of the main thread, it implements concurrent operation between the child threads. Especially in multi-core environment, these threads run concurrently and high rate of encryption and throughput can be obtained, which must effectively raise the operation efficiency of the IDEA-ECB.

**Experiments and Results**

The paper has made a detailed discussion on the design flow of the improved IDEA-ECB. In order to verify the effective feasibility of the improved algorithm, this paper set up a test platform. The source codes are developed by My Eclipse 8.6, and run on a PC with i3-4160 2.2 GHz CPU, 4.0 GB memory and Windows 7 operating system.

In order to realize the IDEA-ECB encryption process based on multi-thread technology, some core function of the algorithm are listed in the following:

```java
private int[] encrypt_Subkey(byte[] byteKey);//52 key
private void encrypt(int[] key, byte[] inbytes, byte[] outbytes);// round of encryption
private byte[] Encrypt(byte[] bytekey, byte[] inputBytes, boolean flag);// enciphered data
public void run();// rewrite the thread body
new Thread().start();// start a thread and enter into the thread state transitions
```

In this paper, the experimental data are generated randomly, with 1 MB, 4 MB, 6 MB, 10 MB, 16 MB data volume as the plaintext input, observe their execution time. And time are achieved by function currentTimeMillis(). Due to the internal state of operation system is in change at any time and it can’t guarantee that the experimental results are consistent every time, the results take the average of execution time of 50 times, and analysis the encryption
rate and throughput of the algorithm. Under the circumstance of identical plaintext input, randomly take 1 MB data volume as the experiment input, then by configuring the number of threads \( \xi \) to 1, 4, 8, 12, 16, observe their execution time, and calculate the encryption speed and throughput. The execution time of generate the cipher text, rate of encryption and throughput for different input data volume are shown in Table 1. The execution time of the cipher text, encryption speed and throughput that randomly take 1MB plaintext as input under the condition of different number of threads are shown in Table 2.

Table 1. Adopt the Classic Encryption Algorithm IDEA- ECB to Generate the Cipher Text.

<table>
<thead>
<tr>
<th>Data Volume (MB)</th>
<th>Time(S)</th>
<th>Encryption Speed (Mbps)</th>
<th>Throughput (M/S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0875</td>
<td>96</td>
<td>11.43</td>
</tr>
<tr>
<td>4</td>
<td>0.3141</td>
<td>107</td>
<td>12.73</td>
</tr>
<tr>
<td>6</td>
<td>0.4384</td>
<td>115</td>
<td>13.69</td>
</tr>
<tr>
<td>10</td>
<td>0.7003</td>
<td>120</td>
<td>14.28</td>
</tr>
<tr>
<td>16</td>
<td>1.1898</td>
<td>113</td>
<td>13.45</td>
</tr>
</tbody>
</table>

Table 2. Adopt the Improved Encryption Algorithm IDEA-ECB with 1 MB Data Volume to Generate the Cipher Text.

<table>
<thead>
<tr>
<th>The Number of Threads ( \xi )</th>
<th>Time(S)</th>
<th>Encryption Speed (Mbps)</th>
<th>Throughput (M/S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0875</td>
<td>96</td>
<td>11.43</td>
</tr>
<tr>
<td>4</td>
<td>0.0509</td>
<td>165</td>
<td>19.64</td>
</tr>
<tr>
<td>8</td>
<td>0.0384</td>
<td>219</td>
<td>26.06</td>
</tr>
<tr>
<td>12</td>
<td>0.0356</td>
<td>236</td>
<td>28.08</td>
</tr>
<tr>
<td>16</td>
<td>0.0214</td>
<td>392</td>
<td>46.65</td>
</tr>
</tbody>
</table>

From Table 1, we can see that, for the classical encryption algorithm IDEA-ECB, the encryption time gradually increase as the increase of the plaintext data volume (as is shown in Fig. 4(a)), encryption speed decrease and throughput is low, and they don’t get effective growth. From Table 2, the experiment adopts the improved encryption algorithm IDEA-ECB with 1MB as the randomly input, by configuring the number of threads to different values. And the results show that the encryption time gradually decrease, the encryption speed increases and the total throughput increases with the increase of the number of threads, the speedup are 1.72, 2.28, 2.46, 2.28, and the throughput of the gradient map is shown in Fig. 4(b).

![Figure 4](image-url)

Figure 4. The Experiment Results of the Improved IDEA-ECB with Multithreaded Encryption.
Experiments have been carried on the comparison and analysis of the encryption speed between the single thread and multithread which $\xi_j$ is randomly set to 8 on different input data volume, as is shown in Fig 5. Obviously, the improved IDEA-ECB algorithm with multithread which $\xi_j$ is randomly set to 8 has obvious differences compared with the single thread. The improved encryption algorithm has higher encryption speed, can be more efficient to protect data and apply in the fields with demanding high encryption speed.

![Figure 5. The Comparison and Analysis of Encryption Rate.](image)

The paper adopt the multithread technology to improve the IDEA-ECB, can maximum send the information which are needed to deal with to the CPU at the same time, obtain a higher encryption and throughput and effectively improve the performance of IDEA-ECB.

**Summary**

On the basis of the theory analysis of classic IDEA-ECB encryption algorithm, this paper changes the order flow of single thread into concurrent flow of multithread with clocking plaintext and multithread technology and proposes an improved IDEA-ECB encryption algorithm. Though the experiment results, they show that the improved algorithm reduces the encryption running time and increases the encryption rate and throughout capacity, satisfying the demand that real time and high effectively software encryption in instant messaging.

**Acknowledgments**

This research is supported by the National Natural Science Foundation of China under Grant No. 61272458, the Natural Science Basic Research Project in Shaanxi Province under Grant No. 2014JM2-6119, and the science and technology plan projects of Yulin city under Grant No. 2014CXY-12.

**References**


