Implementation and Comparison of Two Hash Algorithms

Zhenqi Wang
Information and Network Management Center
North China Electric Power University
Baoding, China
e-mail:xiaoyaobufei@126.com

Lisha Cao
Information and Network Management Center
North China Electric Power University
Baoding, China
e-mail:402293789@qq.com

***Abstract*—MD5 and SHA-1 are the two known Hash Algorithms which are widely applied in information security.They are both developed from MD4.This paper introduces their algorithm logic in detail and their realization using C Language,and compares them through software testing,forms,etc.And finally,we draw a conclusion.**

Keywords-Hash Algorithm; SHA-1; MD5; the realization using C Language; compare

I. INTRODUCTION

Hash algorithm is also called hash function , it plays an important role in modern cryptography[1]. The hash function is a public function, usually denoted as H, we use it to map arbitrary long message to a shorter and fixed-length value, this value is a authentication code, we write it as H(M).The H(M) is also called hash value, hash code ,message digest or array fingerprint. The hash function can be seen as a one-way cryptosystem in the password view, this means that it can only be encrypted, decrypting is impossible. The hash value is a function about the bits of the message, so it provides an error detection capability, because if one bit or a few bits is changed, the hash value will also be changed. In cryptography and data security technologies, the hash function is an important tool that can help us to realize effective, secure and reliable digital signature and certification, it is an important module in security authentication protocol[2].

The most commonly used hash function can be divided into the following categories:

1) Message Digest (MD) Series

MD series is designed by Rivest, a renowned cryptographer and the founder of the public key encryption algorithm RSA, he is also a Turing Award winner. MD series includes MD2 (realized in 1989 for 8 bit computers), MD4 (realized in 1990 for 32-bit computers) and MD5 (proposed in 1991, is an improved version of MD4, it has 128 hash values).

2) SHA (Security Hash Algorithm) series

Based on MD5,In 1993,NIST(National Institute of Standard and Technology) and NSA (National Security Agency) first proposed SHA-0, in 1995 SHA-1 algorithm is proposed (the United States FIPS PUB 180-1 standard), the message hash to 160 values. Today SHA-1 has become the DSA Digital Signature Standard. In 2003, FIPS PUSI80-1 gradually expanded SHA-1 series algorithm, proposing the SHA-256, SHA-384 and SHA-512 algorithm.

3) Other hashing algorithm

HAVAL can be used to achieve the output of the variable hash, RIPEMD-128, RIPEMD-160 are alternative algorithms proposed by the European MD5 and MD4 algorithm researchers. Using Hash algorithm on 64-bit and 32-bit computer very well is the important design thinking of the Tiger algorithm.MD5 and SHA-1 are two international hash function in the field of information security.

II. THE INTRODUCTION AND IMPLEMENTATION OF THE MD5 ALGORITHM

A. *The application of MD5 algorithm*

Its typical application[3] is to generate fingerprint according to a piece of message(byte message) to prevent tampering. For example, using the files to generate unique MD5 message digests through the MD5 algorithm. During the spreading process of the file, regardless of what the file's contents were changed(including human modifying or transmission error, etc.), as long as you recalculate MD5 of this file, you will find information summary is different, it can be sure that you get just an incorrect file. In encryption and decryption technology, the user's password was encrypted by MD5 and stored in file system. When the user logs in, the system transforms password into the MD5 value, and then compare the MD5 value with the value stored in the file system. So although the system does not know the user's password clearly, it can determine the legitimacy of this login. This not only avoids the user's password is known by the user with system administrator privileges, but also increase the difficulty of password is cracked to some extent.

B. *MD5 algorithm logic*

MD5 hash function has two inputs: a group of 512bits' plaintext block, the other group is 128bits' output block (or IV initial variables). The input 128bits' respectively stored in the 32-bit cache: A, B, C, and D. Each 512bits'lock is divided into 16 words (each word 32bit), after four cycles, 16 calculation times of each round, generate 128bits' output after 64 times. The MD5 algorithm's entire structure is shown in Figure 1.

978-0-7695-5004-6/13 $26.00 © 2013 IEEE
DOI 10.1109/ICCIS.2013.195

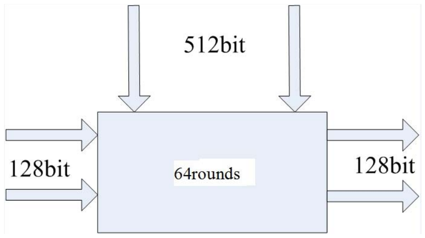


Figure 1. The MD5 algorithm's entire structure

The MD5 algorithm requires five steps, the abstract generation processes are shown in Figure 2.

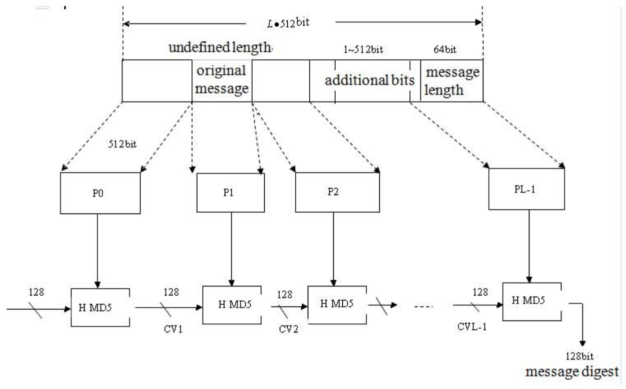


Figure 2. The abstract generation processes

The first step, adding padding bits. Fill the news to make its length can be divided by 512 when added 64. Filling method: fill 1 after the message, and then fill the required numbers of 0. The padding bits can have 1 to 512 numbers.

The second step, adding padding length. Attach 64 bits' representation of the original length of the message to the message which has been filled. When the original message's length is greater than 2^{64}, we fill it with the message's length mod 2^{64}.While The message's length can be divided by 512.,it can be expressed as L data blocks of the 512bits: P_0 , P_1 ,…, P_{L-1} , and its length is L × 512bit.

The third step, initializing MD buffer. We use A 128bits' MD buffer to save the results of the intermediate and final Hash function. It can be expressed as four 32-bit registers (A, B, C, D). The registers are initialized to the following hexadecimal values: A = 67452301; B = EFCDAB89; C = 98BADCFE; D = 10325476.

The fourth step, process input messages at the group of the 512 bits. Process each message block (512 bits = 16 words, each word is 32 bits), each message block can be divided into 16 words: M[0],M[1],…, M[15].This step is the main loop of the MD5, including four loops. The 512 bits that are being processed currently, the 128 bits' buffer value and the ABCD are the inputs of each cycle, after each cycle the buffer's content is updated. Four loop's operation are similar, each one carried out 16 times of operation, but the order of accessing data of each round is different. Each operation does a nonlinear function operation according to the three of a, b, c and d, the results obtained is then coupled with the fourth variable, and the resulting results rightward displacement of an indefinite number , and coupled with one of the a, b, c, and d. Finally, using the result to replace one of the a, b, c and d [4]. Each uses a different basic logic function, referred to F, G, H, I. Among them:

$$F(B,C,D)=(B\wedge C)\vee(\overline{B}\wedge D)$$

$$G(B,C,D)=(B\wedge D)\vee(C\wedge \overline{D})$$

$$H(B,C,D)=B\oplus C\oplus D$$

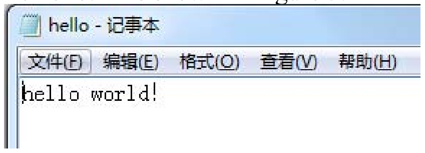
$$I(B,C,D)=C\oplus(B\vee \overline{D})$$

A description of these four functions: If the corresponding bits of B, C and D are independent and uniformly, then every bit of the results is independent and uniformly. F is a function of a bit-by-bit computing. That is, if B, then C, or D. The function H is a bit-wise parity operator.

The fifth step, get the output results. After process all L blocks of 512 bits' data block, according to the output of the four registers A, B, C, D we can get 128 bits' message digest(Starting with the low byte of the A ,ending with the high byte of the D).

C. Use c language to realize the MD5 algorithm

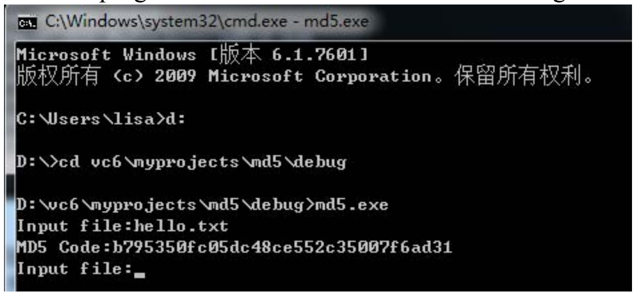
Program first conducted a series of macro definition with parameters, including the four functions F, G, H, I, each round of cycle requires operating FF, GG, HH, II, left shift operation, as well as the fifth step's high and low bit swap operation. Next, define the void md5() function, which is the core of MD5 algorithm. It needs A total of four 64 operations. Enter the main function, open the file using the get function to avoid scanf spaces segmentation data, support for file drag and drop functionality; Use the for loop's method. In the loop first execute the self-defined function md5 (), and then perform function memset to cleared the operation, and then call the fread function, lop until the file's end; implement the first and second procedures of last section, implement the MD5 () function again; at last, the main function get final output results as the fifth step of the previous section. The contents of file hello.txt is shown in Figure 3.



```
hello world!
```

Figure 3. The contents of files

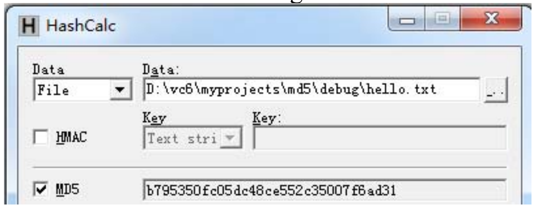
The c program's execution results are shown in Figure 4:



```
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\lisa>d:

D:\>cd vc6\myprojects\md5\debug

D:\vc6\myprojects\md5\debug>md5.exe
Input file:hello.txt
MD5 Code:b795350fc05dc48ce552c35007f6ad31
Input file:_
```

Figure 4. The c program's execution result

Use the MD5 checking software to test the result's validity, the software's testing result is consistent with the program's result. As shown in Figure 5:


HashCalc

Data: File

Data: D:\vc6\myprojects\md5\debug\hello.txt

☐ HMAC Key: Text stri Key:

☑ MD5 b795350fc05dc48ce552c35007f6ad31

Figure 5. Testing result

III. THE INTRODUCTION AND IMPLEMENTATION OF THE SHA-1 ALGORITHM

A. SHA-1 algorithm application

SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512 are all applied by the United States federal government which needs secure hash algorithms. They also use other cryptographic algorithms and protocols to protect sensitive unclassified information. FIPS PUB 180-1 also encouraged personal or commercial organizations to use SHA-1 encryption. Fritz-chip is likely to use the SHA-1 hash function to realize the digital rights' management on the PC [5]. Secure Hash Algorithm is mainly applied to the Digital Signature Algorithm (DSA) which is defined by the Digital Signature Standard. As the MD5 algorithm , SHA-1 algorithm can also be used to verify the integrity of the data and the encrypted message (password, files, etc.). In fact SHA-1 is currently the most widely used worldwide hash algorithm, and has become the standard of the industry.

B. SHA-1 algorithm's logic

The design of SHA-1 algorithm imitates MD4 mostly, which accepts the bits of the maximum length of the message, to generate a 160-bit message digest. Similar with the MD5, this arithmetic operation is divided into 32-bit word of 512 bits' length block for processing units, including four loop operators, 20 rounds per loop, a total of 80 rounds.

The SHA-1 algorithm requires five steps, the abstract's generation processes are shown in Figure 6.

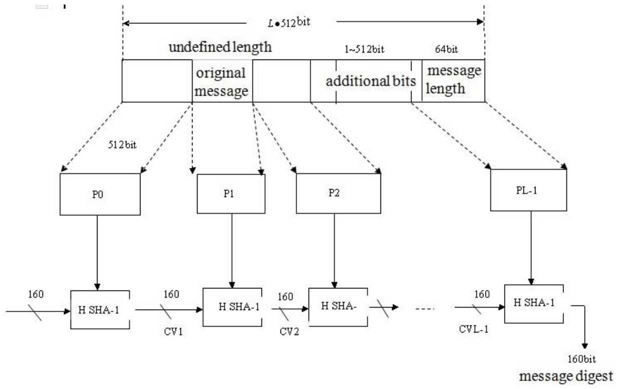


Figure 6. The abstract generation processes

The first step,filling the message. Add some extra bits at the end of the news, it's the same as MD5. So that the length of the message is equal to 448 divided by 512 (448 = 512-64), if the original length is exactly 448 divided by 512, you have to fill an additional 512-bit block. That is to say at least we need to fill a 512 bits' block: the first bit is 1,the others are all 0.

The second step, make up the length. Filling the end of the message with a 64-bit block, the 64-bit is a binary representation of the length of the original message. If the binary representation's bits' numbers are less than 64, the left are filled with 0, so that the block's length is just equal to 64. While if the binary representation's bits' numbers are greater than 64, then only take the 64 bits of the low bytes. Ultimately, the length of the message can be divided by 512.

The third step, buffer's initialization. Decide the initial output of the SHA-1, store the results in five 32-bit registers: A, B, C, D and E, these registers will then be used to maintain the intermediate and final results of the hash function, 5 registers' initial value has 160 bits. The initial variables (hexadecimal):A=67452301, B=EFCDAB89, C=98BADCFE, D=10325476, E C3D2ElF0. The front four values are the same as MD5.

The fourth step, the data processing. The message is divided into 512-bit blocks, each block consists of 16 numbers of 32-bit word, through mixing and shift block, the 16 words is expanded to 80 words and stored in the W[k], k=0，1…，79. Each round's structure is similar, but has different logic function, located respectively as f_1, f_2, f_3, f_4, the input of each round is 512 bits, the output is 160 bits. The logic function of f_{1-4} is defined as follows:

$$f_1 = f(t,B,C,D) = (B \wedge C) \vee (\overline{B} \wedge D),\ 0≤t≤19$$

$$f_2 = f(t,B,C,D) = B \oplus C \oplus D\ ,20≤t≤39$$

$$f_3 = f(t,B,C,D) = (B \wedge C) \vee (B \wedge D) \vee (C \wedge D)\ ,40≤t≤59$$

$$f_4 = f(t,B,C,D) = B \oplus C \oplus D\ ,60≤t≤79$$

Unlike 64 constant T tables used by MD5, SHA-1 algorithm only has four constant values in each round, defined by array as shown in Table 1 below.

TABLE I. VALUES OF K_i IN SHA-1 ALGORITHM

loop	round	Input constant	Value format
1	0≦t≦19	K_1=5A827999	$[2^{30} \times \sqrt{2}]$
2	20≦t≦39	K_2=6ED9EBA1	$[2^{30} \times \sqrt{3}]$
3	40≦t≦59	K_3=8F1BBCDC	$[2^{30} \times \sqrt{5}]$
4	60≦t≦79	K_4=CA62C1D6	$[2^{30} \times \sqrt{10}]$

Fifth step, get output results. After processing all L blocks' data, the last L step we can get a 160-bit message digest H (m).

C. Use c language to realize the SHA-1 algorithm

The program first realizes the left/right shift function of the later algorithm, and establishes a structure which has 5 cache function variables and three other variables:

```
typedef struct {
    u32  h0,h1,h2,h3,h4;
    u32  nblocks;
    unsigned char buf[64];
    int  count;
} SHA1_CONTEXT;
```

Defines the structure type's variable named SHA1_CONTEXT. Then we define a series of custom functions called by main function. Include:

void sha1_init(SHA1_CONTEXT *hd)/*used to calculate the data item initialization of the structure, on section 3.2 , the third step, initializing variables*/

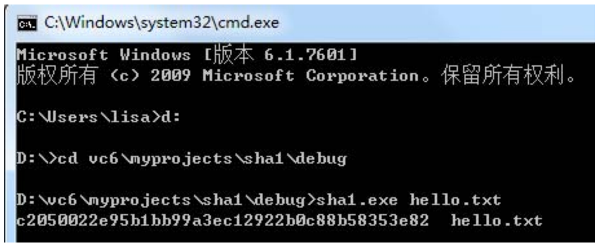
static void transform(SHA1_CONTEXT *hd, unsigned char *data) /* the static function transform is the core function of the SHA-1,it realilzes 80 operations, similar to the MD5 algorithm, before the 80 operations, do a series of macro definitions that will be used*/

static void sha1_write(SHA1_CONTEXT *hd, unsigned char *inbuf, size_t inlen)/* update the message digest with the character variable whose length is inlen*/

static void sha1_final(SHA1_CONTEXT *hd) /*the static function sha1_final is defined to complete the first step in Section 3.2 (call sha1_write function)*/

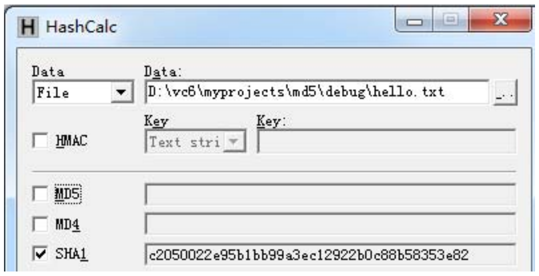
After the program enters the main function, it first do write/read related operations to the fopen 、 fprinf file functions. At last call the function defined before to calculate and output the hash result. Experimental file hello.txt is the same as MD5 test file, so here we do not show the screenshot. The program generates an executable file, the result is shown in Figure 7.

Use the SHA-1 checking software to test the result's validity, the software's testing result is consistent with the program's result. As shown in Figure 8:



```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\lisa>d:

D:\>cd vc6\myprojects\sha1\debug

D:\vc6\myprojects\sha1\debug>sha1.exe hello.txt
c2050022e95b1bb99a3ec12922b0c88b58353e82  hello.txt
```

Figure 7. The result



HashCalc

Data: File
Data: D:\vc6\myprojects\md5\debug\hello.txt
☐ HMAC Key: Text stri Key:
☐ MD5
☐ MD4
☑ SHA1 c2050022e95b1bb99a3ec12922b0c88b58353e82

Figure 8. Testing result

IV. COMPARISON OF THE MD5 ALGORITHM AND THE SHA-1 ALGORITHM

As MD5 and SHA-1 are evolved from MD4, there are many similarities between them, such as the structure and the strength, etc. But there are still differences between them.

A. *Comparison of security*

1) *Collision rate comparison*

If the hash values of the two input strings are the same, then we call the two strings is a collision. We want to make the string of any length into a fixed-length string, so there must be an output string corresponding to multiple input strings, collisions are bound to exist. A concept exists in cryptography called theory crack, refer to propose an algorithm to find collisions at lower theoretical enumeration times [6].

References [7] does several simulation experiments and shows that the SHA-1's CR value is lower than MD5, which means that SHA-1 has higher security then MD5. Shown in Figure 9, Figure 10.

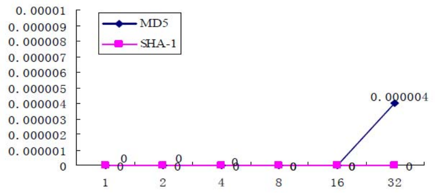


Figure 9. CR of MD5 and SHA-1 (continual/ bit complemented)

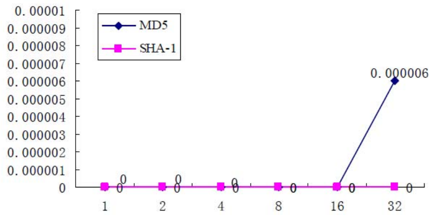
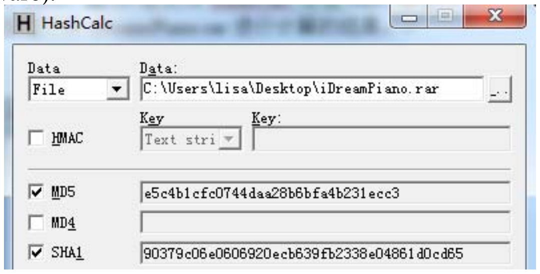


Figure 10. CR of MD5 and SHA-1 (discrete/ bit complemented)

2) *Comparison based on the length of the hash value*

For the same file's hash value, longer value will be harder to crack . The MD5 algorithm's hash value has 128 bits, while SHA-1 algorithm's hash value has 160 bits, the biggest difference between MD5 and SHA-1 is that the MD5's summary is less than SHA-1(32-bit).The 3.85MB compressed file iDreamPiano.rar's calculated results by MD5 and SHA-1are shown in Figure 10(using the HashCalc software).



HashCalc

Data: File
Data: C:\Users\lisa\Desktop\iDreamPiano.rar
☐ HMAC Key: Text stri Key:
☑ MD5 e5c4b1cfc0744daa28b6bfa4b231ecc3
☐ MD4
☑ SHA1 90379c06e0606920ecb639fb2338e04861d0cd65

Figure 11. Comparison of the same file's calculated results using MD5 and SHA-1

3) Comparison of the cracking method

The MD5 algorithm has cracking method, while the SHA-1 is only theoretical cracked, so the MD5 is more vulnerable.

B. Execution speed comparison

Implementing SHA-1 requires 80 steps, while the implementation of the MD5 only requires 64 steps. And compared with the 128-bit buffer of the MD5, SHA-1 must deal with 160 bits of the buffer. Therefore, SHA-1 runs slowly on the same hardware than MD5[8]. The software test results can prove the theory, as shown in Table 2.

TABLE II. THE COMPARISON OF MD5 AND SHA-1EXECUTION TIME

FileName	**FileSize**	**MD5execution time**	**SHA-1execution time**
Test1.exe	18.0MB	180.2592ms	210.3024m
Test2.msi	3.65MB	20.0288m	40.0756m
Test3.doc	58KB	10.0144ms	10.0144ms
Test4.txt	14.4KB	0.01ms	0.01ms

C. Comprehensive comparison

The same points: the MD5 algorithm and SHA-1 algorithms are evolved from MD4 algorithm, they are all hash function. They all need five steps to complete the algorithm, that is filling message, making up the length, initializing variables, data processing and final output. Among them, the first two steps are the same, and the fourth step all requires four operations, the message is divided into a plurality of 16-word (32 bits) i.e. 512-bit message block to handle. Therefore, their structures and algorithms have many similarities.

Different points: In order to facilitate comparison, the two hash algorithms are listed to do comprehensive comparison, as shown in Table 3.

TABLE III. COMPREHENSIVE COMPARISON OF MD5 AND SHA-1

	MD5	**SHA-1**
Length of Hash value	128bit	160bit
steps	64（16×4）	80（20×4）
The max length of message	Not limited	≤ 2^{64} bit
Nonlinear function（basic logic function）	4	3（2、4round is the same）
Constant numbers	64	4
structure	Little-endian format	Big-endian format
security（relatively）	lower	higher
Execution rate（the same hardware，relatively）	faster	slower

V. CONCLUSION

In this paper, we introduced two hash algorithms: MD5 algorithm and SHA-1 algorithm's logical procedures and their realization by c language. We conclude their different points and similarities through two hash algorithm's study. The biggest difference between them is that MD5 algorithm's message digest is 32-bit longer than SHA-1 algorithm, which makes the SHA-1 algorithm more secure than MD5 algorithm. However, due to the different steps in the core part of the algorithm, SHA-1 algorithm requires 80 steps, MD5 algorithm only requires 64 steps. And the SHA-1 algorithm has 5 registers of total 160 bits, MD5 algorithm has four registers of total 128 bits. So on the same hardware, SHA-1 run slower then MD5. To a certain extent, we can conclude that, when the processing data is very small, we can choose to use the MD5's message digest rather than SHA-1's message digest. Because under this condition, the MD5's collision rate and run time are all not high, which can get significant savings in resources.

In modern cryptography, Hash functions initially were used for data integrity, including digital signatures, as well as the applications of digital fingerprint linked with the digital signatures[9]. Because of the hash function's one-way and randomness characteristics, they were not only applied to keep the data integrity, but also often used in cryptography's other areas, for example, the proof, key derivation and pseudo-random numbers' generation, etc.

REFERENCES

[1] Shibin Zhang,Wunan Wan,Jinquan Zhang, "Applied Cryptography," Xi'an:XidianUniversity Press,2009

[2] Jianwei Hu, "Network Security and Confidentiality , " XidianUniversity Press,2003.

[3] The Brief Introduction and Algorithm Source Code of GAME-LAB.MD5, http://www.cnblogs.com/mywolrd/archive/2008/09/18/1930721.html, 2008 .

[4] Xiaoling Wei, "MD5 Encryption Algorithm and Application," Yan'an University Computing Center,2010.

[5] http://baike.baidu.com/view/94209.htm.

[6] Hongjun Liu, "Research and Implementation of MD5 Collision Avoidance and Exhaustive Transform Algorithm,"Baotou,2008.

[7] Ming Hu,Yan Wang, "The Collision Rate Tests of Two Known Message Digest Algorithms," 2009 International Conference on Computational Intelligence and Security,2009,pp.319-323.

[8] S. H. Khayal, A. Khan, N. Bibi and T. Ashraf, "Analysis of Password Login Phishing Based Protocols for Security Improvements,"2009 International Conference on Emerging Technologies,2009.

[9] Yanbo Wang,,"Applied Cryptography," Machinery Industry Press,200