Discussion of One Improved Hash Algorithm Based on MD5 and SHA1[1]

Xiaorui Chan, Guangzhong Liu

Abstract: **We have devised one 160 bits improved hash algorithm based on MD5 and SHA1. And in additional, we also introduce four assistant functions named: F(X,Y,Z), G(X,Y,Z), H(X,Y,Z), and I(X,Y,Z) to do 4 rounds of 16 steps iterative operation with 32-bit data as input and 32-bit as output, saving in A, B, C, D respectively. Then we use one new extending function K(X,Y,Z) to expand 32-bit A, B, C, D to 40-bit. At last, by combining the result from low-bit AA, we realize the 160-bit hash algorithm. By analysis, we have found that: without increasing the time complexity, the improved algorithm has increased the security better compared with MD5 and SHA1 algorithms**.

Keywords: Extending Function, Hash Algorithm, Time Complexity

1 Introduction

Under the incessantly developing society, web security has become one of the most important problems in nowadays. Commerce data commuting between companies also become more confidential. One message losing means one business missing, which might result in a severe damage to a company. Meanwhile, message security also plays a significant role in military affairs, national defenses, and foreign policies.

There are two main kinds of encryption algorithms: Symmetric Key Algorithm (SKA) and Asymmetric Key Algorithm (AKA) [2]. In additional, there is also another assistant algorithm called: Hash, which compresses message of any length to certain fixed length (message-digest). This process is irreversible. Hash function can be used in many fields such as: digital signature, message integrality test, and message originality etc. Hash algorithm mainly includes: MD×(Message－Digest Algorithm), SHA×(Secure Hash Algorithms), N-Hash, RIPE-MD, and HAVAL etc., among which MD5 is the most famous one in nowadays.

However, August 2004, an international cryptogram conference in California America, Professor Wang Xiaoyun [5], from Shandong university, has proclaimed that she has already deciphered MD5, HAVAL-128, MD4, and RIPEMD, provoking a great disturbance in cryptogram field. MD5 and SHA1 were once considered as the most secure algorithms, but through using "Mod Difference" produced by Pro. Wang, we can find "Collide" result of MD5 on common computers only in two hours. And the kernel principle of that method is how to choose an advanced cryptogram algorithm to keep the key secretly.

Under this circumstance, we have to bring out one new 160-bit hash algorithm based on MD5 and SHA1. In order to better compare, we here introduce MD5 and SHA1 concisely first.

1.1 MD5 Algorithm [4]

(1) Data Filling
MD5 algorithm adds supplement right following the input data, making the whole length mod 512 equals 448, namely, extending data to K*512+448 bits, or K*64+56 bytes. K is integer here.

(2) Add Data Length
We use 64-bit data *b* to denote original length of data, and separate *b* to two 32-bit blocks, When $b>2^{64}$, we extend the length to times of 512, which means that the length becomes times of 16 double bytes (32 bits). Save data in array M[0…N-1]. Here N is times of 16.

(3) Initialize Variables
Define four 32-bit variables named A, B, C, and D respectively. Initialize them as follows: A=0x01234567, B=0x89abcdef, C=0xfedcba98, D=0x76543210.

(4) Data Processing
Define four assistant functions:

$$F(X,Y,Z)=(X\wedge Y)\vee(\neg X)\wedge Z)$$
$$G(X,Y,Z)=(X\wedge Z)\vee(Y\wedge(\neg Z))$$
$$H(X,Y,Z)=X\oplus Y\oplus Z$$
$$I(X,Y,Z)=Y\oplus(X\vee(\neg Z))$$

Here, $\wedge$ means 'or', means 'and', $\neg$ means 'not', and $\oplus$ means 'XOR'.
We also have to introduce a table T[i], here i=1…64, and T[i]=int(abs(sin i)^{4294967296}). By using Sine function and Power function, we can erase the linear mistake efficiently.

(5) Final Result
The result in outputted as following sequence: A, B, C, D.

1.2 SHA1 Algorithm [4]

[1] This paper is supported by the grand project of the Science and Technology Commission of Shanghai Municipality (No.06DZ11202), Shanghai Leading academic Discipline Project(No.T0602), and the Subject Foundation of Shanghai Maritime University(No.XL0101-1).

Xiaorui Chan (1982-), Master, College of Information Engineering, Shanghai Maritime University, interested in network security and database. E-mail: sherrychan@163.com.

Guangzhong Liu (1962-), PhD, Professor, College of Information Engineering, Shanghai Maritime University, interested in agent technology, distributed database, and network security..E-mail: gzhliu@cie.shmtu.edu.cn.

SHA1 also is another main hash algorithm, which is primarily based on MD4 principle. Because it produces 160-bit output, so SHA1 needs five 32-bit registers. But the method of message digest and data filling works just like MD5 algorithm. SHA has 4 main rounds iterative, and each round has 20 steps operations. There are only some minute differences including: rotate left, addition constant, and non-linear function. Here are the five initialized variables:

A＝0x67452301
B＝0xefcdab89
C＝0x98badcfe
D＝0x10325476
E＝0xc3d2e1f0

The constants used in process are stated as follows:

$$K_t = \begin{cases} 5a827999 & 0 \leqq t \leqq 19 \\ 6ed9eba1 & 20 \leqq t \leqq 39 \\ 8f1bbcdc & 40 \leqq t \leqq 59 \\ Ca62c1d6 & 60 \leqq t \leqq 79 \end{cases}$$

We can use these constants to judge whether one procedure has adopted SHA1 algorithm. After finishing initialization, we then start the main rotation of algorithm.

2 One Improved Hash Algorithm

Although we have extended this hash algorithm to 160-bit, it is primarily based on MD5, only introducing one excellent assistant function from 160-bit SHA1. So far, we name it MD5plus algorithm temporarily. The process works as following:

i. Information filling module
ii. Initialization module
iii. Hash value calculation module
iv. Bit extending module
v. Output module

2.1 Data Filling

The data filling of MD5plus works almost as the same as MD5 and SHA1. See figure 1.

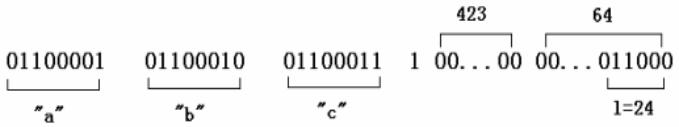

Figure1 Information Filling

We have to append ox80 to low-bit of the handling information (actually, it equals appending some "0" after adding a "1" to the information). Supply the information with "0" until length mod 512 equals to 448. At last, append information length in the end (most hash algorithms use this method to fill data length).

2.2 Initialize Hash Value

Create four 32-bit buffers named A, B, C, D and four 40-bit buffers named AA, BB, CC, DD. We here have to choose the initialization of MD5 as the initialized value of A, B, C and D in MD5plus algorithm, because the five values in SHA1 are not suitable.

A=0X01234567
B=0X89abcdef
C=0Xfedcba98
D=0X76543210

But here, we still initialize AA, BB, CC, and DD with all "0".

2.3 Operation Functions

In order to consolidate the security of MD5plus, we should bring in four assistant functions firstly. With 32-bit input and 32-bit output, each function has three parameters,

$$\begin{cases} F(X,Y,Z)=(X \wedge Y) \vee (\neg X) \wedge Z) & \ldots\ldots\ldots\ldots(1) \\ G(X,Y,Z)= X \oplus Y \oplus Z & \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2) \\ H(X,Y,Z)=(X \wedge Z) \vee (Y \wedge (\neg Z)) & \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(3) \\ I(X,Y,Z)= X \oplus Y \oplus Z & \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(4) \end{cases}$$

All the process includes four rounds, and each round contains 16 steps. Here are the concrete operational functions:

FF(a,b,c,d,Mj,s,ti) means: a=b+((a+(F(b,c,d)+Mj+t[i])<<<s)
GG(a,b,c,d,Mj,s,ti) means: a=b+((a+(G(b,c,d)+Mj+t[i])<<<s)
HH(a,b,c,d,Mj,s,ti) means: a=b+((a+(H(b,c,d)+Mj+t[i])<<<s)
II(a,b,c,d,Mj,s,ti) means: a=b+((a+(I(b,c,d)+Mj+t[i])<<<s)

Mj means the *j* block in whole information data, <<< means rotate left, t[1…64] means an array of 64 constants, and still, $t[i]=int(abs(\sin i)^{4294967296}$, the unit *i* is radian.

Here is the process in detail.

```
For i=0 to N/16-1 do
For j=0 to 15 do
Set X[j] to M[i*16+j];
End;
a=A;    //Save A as a
b=B;    //Save B as b
c=C;    //Save C as c
d=D;    //Save D as d
```

Do the following 64 operations:

The first round:
FF(a, b, c, d, M0, 7, 0xd76aa478)
FF(d, a, b, c, M1, 12, 0xe8c7b756)
FF(c, d, a, b, M2, 17, 0x242070db)
FF(b, c, d, a, M3, 22, 0xc1bdceee)
FF(a, b, c, d, M4, 7, 0xf57c0faf)
FF(d, a, b, c, M5, 12, 0x4787c62a)
FF(c, d, a, b, M6, 17, 0xa8304613)
FF(b, c, d, a, M7, 22, 0xfd469501)
FF(a, b, c, d, M8, 7, 0x698098d8)
FF(d, a, b, c, M9, 12, 0x8b44f7af)
FF(c, d, a, b, M10, 17, 0xffff5bb1)
FF(b, c, d, a, M11, 22, 0x895cd7be)
FF(a, b, c, d, M12, 7, 0x6b901122)
FF(d, a, b, c, M13, 12, 0xfd987193)
FF(c, d, a, b, M14, 17, 0xa679438e)
FF(b, c, d, a, M15, 22, 0x49b40821)

The second round:
GG(a, b, c, d, M1, 5, 0xf61e2562)
GG(d, a, b, c, M6, 9, 0xc040b340)
GG(c, d, a, b, M11, 14, 0x265e5a51)
GG(b, c, d, a, M0, 20, 0xe9b6c7aa)
GG(a, b, c, d, M5, 5, 0xd62f105d)
GG(d, a, b, c, M10, 9, 0x02441453)
GG(c, d, a, b, M15, 14, 0xd8a1e361)
GG(b, c, d, a, M4, 20, 0xd8a1e681)
GG(a, b, c, d, M9, 5, 0xe7d3fbc8)
GG(d, a, b, c, M14, 9, 0x21e1cde6)
GG(c, d, a, b, M3, 14, 0xf4d50d87)
GG(b, c, d, a, M8, 20, 0x455a14ed)
GG(a, b, c, d, M13, 5, 0xa9e3e905)
GG(d, a, b, c, M2, 9, 0xfcefa3f8)
GG(c, d, a, b, M7, 14, 0x676f02d9)
GG(b, c, d, a, M12, 20, 0x8d2a4c8a)

The third round:
HH(a, b, c, d, M5, 4, 0xfffa3942)
HH(d, a, b, c, M8, 11, 0x8771f681)
HH(c, d, a, b, M11, 16,0x6d9d6122)
HH(b, c, d, a, M14, 23, 0xfde5380c)
HH(a, b, c, d, M1, 4, 0xa4beea44)
HH(d, a, b, c, M4, 11, 0xbdecfa9)
HH(c, d, a, b, M7, 16, 0xf6bb4b60)
HH(b, c, d, a, M10, 23, 0xbebfbc70)
HH(a, b, c, d, M13, 4, 0x289b7ec6)
HH(d, a, b, c, M0, 11, 0xeaa127fa)
HH(c, d, a, b, M3, 16, 0xd4ef3085)
HH(b, c, d, a, M6, 23, 0x04881d05)
HH(a, b, c, d, M9, 4, 0xd9d4d039)
HH(d, a, b, c, M12, 11, 0xe6db99e5)
HH(c, d, a, b, M15, 16, 0x1fa27cf8)
HH(b, c, d, a, M2, 23, 0xc7ac5665)

The fourth round:
II(a, b, c, d, M0, 6, 0xf4292244)
II(d, a, b, c, M7, 10, 0x432aff97)
II(c, d, a, b, M14, 15, 0xab9423a7)
II(b, c, d, a, M5, 21, 0xfc93a039)
II(a, b, c, d, M12, 6, 0x655b59c3)
II(d, a, b, c, M3, 10, 0x8f0ccc92)
II(c, d, a, b, M10, 15, 0xffeff47d)
II(b, c, d, a, M1, 21, 0x85845dd1)
II(a, b, c, d, M8, 6, 0x6fa87e4f)
II(d, a, b, c, M15, 10, 0xfe2ce6e0)
II(c, d, a, b, M6, 15, 0xa3014314)
II(b, c, d, a, M13, 21, 0x4e0811a1)
II(a, b, c, d, M4, 6, 0xf7537e82)
II(d, a, b, c, M11, 10, 0xbd3af235)
II(c, d, a, b, M2, 15, 0x2ad7d2bb)
II(b, c, d, a, M9, 21, 0xeb86d391)

What's more, we need to define another special extending function: K(X,Y,Z)=(X∧Y) ∨(X∧Z) ∨(Y∧Z). With 40-bit input and output, it is the kernel of MD5plus. Here is the main principle of MD5plus. See figure 2.

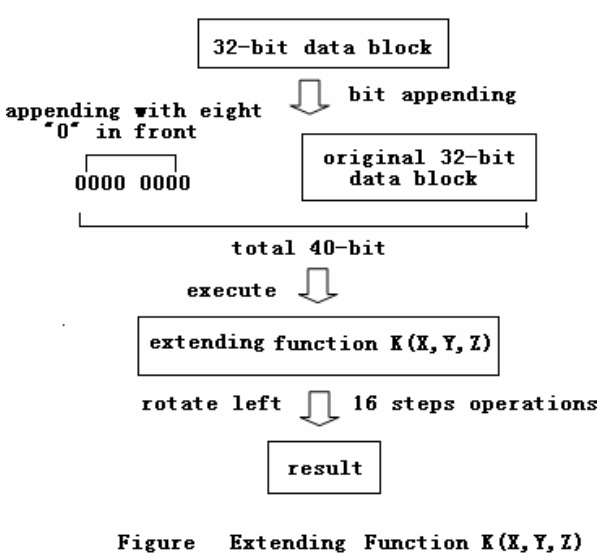

Figure Extending Function K(X,Y,Z)

After finishing former 64 steps operations, we only have to append the results with eight "0" in front, then, save them to 40-bit registers AA, BB, CC, and DD respectively. In addition, by running 16 steps of kernel appending function K(X,Y,Z), we get four 40-bit outputs. At last, adopting cutting and combination method [1], we get the 160-bit hash result.

And the 16 steps operation in this round work just like the former four rounds. Here is the concrete processing:

KK(a,b,c,d,Mj,s,ti) means: a=b+((a+(K(b,c,d)+Mj+t[i])<<<s)

2.4 Result

Here is the final format of output: AA, BB, CC, and DD.

3 Performance Comparison

MD5plus algorithm based on MD5, and absorbed some excellent functions from SHA1. In hash length, MD5plus has improved to 160-bit, compared with 128-bit MD5. It also means that MD5plus algorithm has increased 25% in security length, avoiding the deadly defect of MD5. After adding assistant extending function K(X,Y,Z), MD5plus needs a slice of more time then MD5 in Computing Time Spending (CTS), but, in the long run, MD5plus algorithm works better as a whole.

And compared with SHA1 algorithm, MD5plus avoided the defect of initialization of five hash values at beginning. It seems that we only have reduced one parameter, however, in machine computing, we actually have cut down 4 rounds of 16 steps (16*4=64) computing time, decreasing CTS efficiently. Instead, MD5plus adopted an easy, ingenious method. We only have to add an extending function K(X,Y,Z) in the later period of the task. Although by increasing 16 steps extra iterative operations seems cost more CTS, in fact, MD5plus perfectly avoided 64 steps computer operations in the beginning, saving plenty of precious CTS.

We all have already known that the collision complexity of one couple in 128-bit MD5 is 2^{64}, and the collision complexity of 160-bit SHA1 is 2^{80} (After deciphered MD5, Professor Wang Xiaoyun proclaimed that the new collision complexity of SHA1 is 2^{69}, not 2^{80}). Through the principle process of MD5plus, we can easily figure out that the collision complexity of MD5plus is $2^{64}+2^{16}$, but 2^{16} is far less then 2^{64}. So the final result of MD5plus algorithm is $2^{64}+2^{16}\approx 2^{64}$.

Here is the comparison among MD5, SHA1, and MD5plus algorithm. See Table 1.

Table 1: Comparison among MD5, SHA1, and MD5plus

Funciton	MD5	SHA1	MD5plus
Block length	512 bit	512 bit	512 bit
Algorithm length	128 bit	160 bit	160 bit
Rotation steps	64 steps	80 steps	80 steps
Initialization variables	4	5	4
Collision complexity	2^{64}	2^{80}	$2^{64}+2^{16}\approx 2^{64}$

4 Conclusion

MD5plus algorithm has palpable advantages not only in security, but also in computing time spending. Besides, combining with other 160-bit asymmetric algorithms, we could use MD5plus to compose an excellent plaintext protection system, which will be discussed in detail in my thesis.

Because of the equipment limit, we still confront some difficult in the realization of MD5plus algorithm. So the research we discussed above is still resting on theory level, and how to support this improved algorithm with particular data is what should be further studied in the following research.

Encryption technology cannot solve all the web security problems. And it should be combined to other kinds of means such as: macroscopic managements, law, policy, and education etc.

References

[1] Xiaorui Chan, Guangzhong Liu, *Discussion of One New Symmetric Algorithm*. The International MultiConference of Engineers and Computer Scientists, 2007
[2] J. Richter, *Programming Applications for Microsoft Windows*. Microsoft Press, 2000
[3] J. Robbins, *Debugging Applications*. Microsoft Press, 1999.
[4] D. Guo, F. Sun, Z.M. Tang, *Encryption and Deciphering,* Tsinghua University Press, 2000.
[5] Z.Y. Hu, *Password Breaking and Encryption Technology.* Machine Industry Press, 1999.
[6] X.S. Lai, L. Han, Z.C. Zhang. *Computer Cryptography and Application Cryptography.* National Defense Industry Press, 2001