iranphp articles

عنوان مقاله :	MD5 چیست؟
نگارنده :	امید متقي راد
آدرس پست الکترونیك :	omid@oxygenws.com
تاریخ نگارش :	.....

MD5 چیست؟

۱-خلاصه :

در این مقاله با الگوریتم "خلاصه پیام "MD5 آشنا می شویم. این الگوریتم یک رشته با طول متفاوت را به عنوان ورودی می گیرد و یک "خلاصه پیام "MD5 یا "اثر انگشت" با طول 128 بیت می سازد .

در این روش اینکه دو پیام مختلف دارای یک "خلاصه پیام" باشند یا اینکه یک رشته از روی یک "خلاصه پیام" ساخته شود غیر ممکن می باشد. این الگوریتم برای امضاهای دیجیتال مناسب است، جایی که احتیاج به خلاصه کردن یک فایل بزرگ در یک رشتهء امن و فشرده، قبل از کد کردن آن متن، در سیستم های کدینگ، با کلید های خصوصی و عمومی آن سیستم مانند RSA (Rivest Shamir Adelman)

الگوریتم MD5 برای داشتن سرعت بالا در ماشین های ۳۲ بیتی طراحی شده است در عین حال احتیاجی به جانشینی ها در جداول بزرگ ندارد. این الگوریتم را با کدهای بسیار کمی می توان نوشت .

الگوریتم MD5 توسعه ای از الگوریتم MD4 می باشد با این تفاوت که MD5 کمی کندتر از MD4 عمل میکند اما در طراحی آن بسیار محافظه کارانه عمل شده است .

MD5 به این دلیل طراحی شد که حس کردند MD4 به عنوان سرعت بالایی که داشت پذیرفته شده و از امنیت بالایی در شرایط بحرانی برخوردار نمی باشد . MD4 برای سرعت بالا طراحی شده ولی احتمال شکست آن در رمز کردنی موفق وجود دارد MD5 . کمی در سرعت کند شده با dl‌n تفاوت که بیشترین امنیت را داراست. این الگوریتم حاصل تاثیر دادن نظرات تعدادی از استفاده کنندگان MD4 به همراه مقادیری تغییر در ساختار الگوریتم برای افزایش سرعت و قدرت آن می باشد. الگوریتم MD5 در این مکان عمومی قرارگرفته تا از آن استفاده و در صورت امکان استاندارد شود .

۲-شرایط و نکات لازم :

در این متن منظور از « کلمه» تعداد ۳۲ بیت و «بایت» تعداد ۸ بیت داده می باشد. یک صف از بیت ها دارای خصوصیات طبیعی یک صف از بایتها می باشند که هر گروه هشت تایی متوالی از بیتها یک بایت را تشکیل می دهند که پرارزش ترین بیت در ابتدا قرار دارد . یک صف از بایت ها دقیقا مشابه یک صف ۳۲ بیتی از کلمات پردازش می شود. جایی که گروهی 4 تایی از توالی بایتها پردازش می شوند، کم ارزش ترین بایت اولین بایت می باشد .

اجازه بدهید از x_i بجای x) xi اندیس (i استفاده کنیم و اگر مقدار اندیس یک عبارت محاسباتی بود آن را در {} محدود می کنیم، مانند : x_{i-1}همچنین از ^ به عنوان علامت توان استفاده می کنیم، پس x^i یعنی xبه توان i

اجازه بدهید از علامت «+» برای اضافه کردن دو کلمه به هم استفاده کنیم. از x<<<5 به عنوان عملگر چرخش بیتی در کلمات استفاده می شود که xبه اندازه ۵ بیت به چپ چرخش می کند .

از not (x) به عنوان عملگر نقیض بیتی، از X v Y به عنوان عملگر فصل (or) و از X xor Y به عنوان عملگر exclusive or و از XYبه عنوان عملگر عطف (and) استفاده می کنیم .

۳- توضیحات الگوریتمMD5 :

فرض کنید ما b بیت پیام به عنوان ورودی داریم و تصمیم داریم خلاصه پیام آن را بدست آوریم b . در ایینجا یک عدد نا منفی و صحیح است، b می تواند مقدار صفر داشته باشد و هیچ محدودیتی برای مضرب هشت بودن آن نیست و به هر اندازه می تواند بزرگ باشد. فرض کنید بیت های این پیام را بشود به صورت زیر نوشت :

```
m_0 m_1 ... m_{b-1}
```

برای آوردن خلاصه پیام ۵ مرحله زیر را انجام می دهیم :

گام ۱- اضافه کردن بیتهای نرم کننده :

طول پیام مورد نظر به ۴۴۸ به پیمانه ۵۱۲ توسعه پیدا می کند به این معنی که اگر به طول پیام ۶۴ بیت اضافه شود، طولش مضربی از ۵۱۲ خواهد بود. عمل توسعه دادن همیشه اجرا می شود مگر اینکه طول پیام به صورت ۴۴۸ به پیمانه ۵۱۲ باشد .

عمل توسعه پیام یا نرم کردن آن به صورت زیر انجام می شود :

یک بیت [۱] سپس تعدادی بیت [۰] به پیام اضافه می شود.اضافه شدن بیت های ۰ تا زمانی که طول رشته به ۴۴۸ بر پایه ۵۱۲ برسد، ادامه پیدا می کند. در این عمل حداقل یک بیت و حداکثر ۵۱۲ بیت اضافه خواهد شد .

گام ۲- افزایش طول :

یک نمایش ۶۴ بیتی از b بیت پیام اولیه به آخر نتیجه گام قبل اضافه می شود. در بدترین حالت، bبزرگتر از ۶۴ بیت خواهد بود. در این حالت فقط ۶۴ بیت کم ارزش b استفاده خواهد شد .

هم اکنون طول پیام بدست آمده دقیقا معادل مضربی از ۵۱۲ خواهد بود. مشابه اینکه بگوییم، این پیام طولی معادل مضربی از ۱۶ کلمه دارد اجازه بدهید M[0...N-1]را نمایانگر کلمات پیام بدست آمده بدانیم . (Nمضربی از ۱۶ می باشد)

گام ۳- بین بافر برایMD :

برای محاسبه خلاصه پیام یک بافر ۴ کلمه ای (A,B,C,D) استفاده می شود. هر کدام ازA، B ، Cو D یک ثبات ۳۲ بیتی می باشند. این ثبات ها مطابق جدول زیر مقدار دهی می شوند (بایتهای کم ارزش در ابتدا قرار دارند)

```
word A: 01 23 45 67
word B: 89 ab cd ef
word C: fe dc ba 98
word D: 76 54 32 10
```

گام ۴- پردازش پیام در بلاک های ۱۶ کلمه ای :

در ابتدا ۴ تابع کمکی تعریف می کنیم که هر کدام به عنوان ورودی سه کلمهء ۳۲ بیتی می گیرد و برای خروجی یک کلمهء ۳۲ بیتی تولید می کند .

```
F(X,Y,Z) = XY v not(X) Z
G(X,Y,Z) = XZ v Y not(Z)
H(X,Y,Z) = X xor Y xor Z
I(X,Y,Z) = Y xor (X v not(Z))
```

در هر موقعیت بیتی، Fبه عنوان شرط عمل می کند:اگر X آنگاه Y در غیر این صورت . Z تابع F می توانست طوری تعریف شود که به جای استفاده از vاز + استفاده کند چون XY و not(X) هرگز یک هایی در موقعیت بیتی یکسان نخواهد داشت. جالب است به یاد داشته باشید که اگر بیت هایX ، Yو Zمستقل و غیر مرتبط باشند، هر بیت از F(X, Y, Z) مستقل و غیر مرتبط خواهد بود .

توابعG ، H وI شبیه تابع F هستند، به طوری که آنها در "توازی بیتی" کار می کنند تا خروجی شان را از بیت هایX ، Yو Z تولید کنند .در چنین روشی اگر بیت های متناظرX ، Yو Z مستقل و غیر مرتبط باشند، آنگاه هر بیت از G(X, Y, Z) ، H(X, Y, Z)وI(X, Y, Z) مستقل و غیر مرتبط خواهند بود .

توجه داشته باشید که تابعH ، تابع XOR یا توازن بیتی از ورودی هایش است. این گام از یک جدول ۶۴ عنصری T[1...64] ساخته شده از یک تابع مثلثاتی، استفاده می کند. اجازه دهید T[i] ، i-امین عنصر جدول را مشخص می کند که برابر است با قسمت صحیح حاصلضرب 4294967296 در abs(sin(i))، به طوری که I به رادیان باشد .

کارهای زیر را انجام می دهید :

```
/* Process each 16-word block. */
   For i = 0 to N/16-1 do

     /* Copy block i into X. */
     For j = 0 to 15 do
       Set X[j] to M[i*16+j].
     end /* of loop on j */

     /* Save A as AA, B as BB, C as CC, and D as DD. */
     AA = A
     BB = B
     CC = C
```

```
    DD = D

    /* Round 1. */
    /* Let [abcd k s i] denote the operation
         a = b + ((a + F(b,c,d) + X[k] + T[i]) <<< s). */
    /* Do the following 16 operations. */
    [ABCD  0  7  1]  [DABC  1 12  2]  [CDAB  2 17  3]  [BCDA  3 22  4]
    [ABCD  4  7  5]  [DABC  5 12  6]  [CDAB  6 17  7]  [BCDA  7 22  8]
    [ABCD  8  7  9]  [DABC  9 12 10]  [CDAB 10 17 11]  [BCDA 11 22 12]
    [ABCD 12  7 13]  [DABC 13 12 14]  [CDAB 14 17 15]  [BCDA 15 22 16]

    /* Round 2. */
    /* Let [abcd k s i] denote the operation
         a = b + ((a + G(b,c,d) + X[k] + T[i]) <<< s). */
    /* Do the following 16 operations. */
    [ABCD  1  5 17]  [DABC  6  9 18]  [CDAB 11 14 19]  [BCDA  0 20 20]
    [ABCD  5  5 21]  [DABC 10  9 22]  [CDAB 15 14 23]  [BCDA  4 20 24]
    [ABCD  9  5 25]  [DABC 14  9 26]  [CDAB  3 14 27]  [BCDA  8 20 28]
    [ABCD 13  5 29]  [DABC  2  9 30]  [CDAB  7 14 31]  [BCDA 12 20 32]

    /* Round 3. */
    /* Let [abcd k s t] denote the operation
         a = b + ((a + H(b,c,d) + X[k] + T[i]) <<< s). */
    /* Do the following 16 operations. */
    [ABCD  5  4 33]  [DABC  8 11 34]  [CDAB 11 16 35]  [BCDA 14 23 36]
    [ABCD  1  4 37]  [DABC  4 11 38]  [CDAB  7 16 39]  [BCDA 10 23 40]
    [ABCD 13  4 41]  [DABC  0 11 42]  [CDAB  3 16 43]  [BCDA  6 23 44]
    [ABCD  9  4 45]  [DABC 12 11 46]  [CDAB 15 16 47]  [BCDA  2 23 48]

    /* Round 4. */
    /* Let [abcd k s t] denote the operation
         a = b + ((a + I(b,c,d) + X[k] + T[i]) <<< s). */
    /* Do the following 16 operations. */
    [ABCD  0  6 49]  [DABC  7 10 50]  [CDAB 14 15 51]  [BCDA  5 21 52]
    [ABCD 12  6 53]  [DABC  3 10 54]  [CDAB 10 15 55]  [BCDA  1 21 56]
    [ABCD  8  6 57]  [DABC 15 10 58]  [CDAB  6 15 59]  [BCDA 13 21 60]
    [ABCD  4  6 61]  [DABC 11 10 62]  [CDAB  2 15 63]  [BCDA  9 21 64]

    /* Then perform the following additions. (That is increment each
       of the four registers by the value it had before this block
       was started.) */
    A = A + AA
    B = B + BB
    C = C + CC
    D = D + DD

end /* of loop on i */
```

گام ۵ - خروجی :

خلاصه پیامی که به عنوان خروجی تولید می شود و عبارت است از A ، B، C وD ، که ما با کم ارزش ترین بیت A شروع می کنیم و به با ارزش ترین بیت D خاتمه می دهیم. این تعریف MD5 را کامل می کند .

۴-نتیجه :

الگوریتم خلاصه پیام MD5 به سادگی قابل اجرا می باشد و یک "اثر انگشت" یا "خلاصه پیام" از پیام با طول اختیاری تولید می کند. گمان برده مـی شـود کـه امکان مواجه شدن با دو پیام که خلاصه پیام مشابهی دارند از رتبهء ۲^۶۴ و برای هر پیامی که به آن یک خلاصه پیام داده شده است از رتبهء ۲^۱۲۸ می باشد .

الگوریتم MD5 برای نقاط ضعف به دقت بررسی شده است. به هر حال این الگوریتم نسبتا جدید است و تحلیل امنیتی بیشتری را طلب می کند، مشابه طـرح های مشابه در این رده .

۵-پانویس :

این مقاله اطلاعاتی برای جامعهء اینترنتی مهیا کرده و البته هیچ استانداردی را مشخص نمی کند. انتشار این مقاله به هر تعداد آزاد می باشد .

بعضی از بخش های این RFC که از اهمیت کمتری برخوردار بود حذف شده .

می تونید فایل doc رو از اینجا بگیرید:

http://www.oxygenws.com/md5.doc