هشتمین کنفرانس شبکه های توزیع نیروی برق
انجمن مهندسین برق و الکترونیک ایران
۳۰ و ۳۱ اردیبهشت ۱۳۸۲

تصدیق اطلاعات در سیستم اتوماسیون توزیع با استفاده از تابع در هم MD5

مرتضی اسماعیلی [۱] فرامرز فقیهی [۲]
۱-استادیار دانشکده ریاضی دانشگاه صنعتی اصفهان
۲-دانشجوی دوره دکتری برق قدرت دانشگاه علم و صنعت ایران / شرکت صنعتی مهرآباد (صنایع آب و برق وزارت نیرو)

چکیده:

گسترش روز افزون شبکه های توزیع سبب شده است تا امکان استفاده از روشهای سنتی نگهداری، بهره برداری و حفاظت شبکه میسر نباشد، لذا برداشت اطلاعات شبکه های توزیع، مدون سازی آنها و بهره گیری از سیستم اتوماسیون امری بدیهی و اجتناب ناپذیر می‌باشد. نرم افزار سیستم اتوماسیون توزیع باید دارای قابلیتهای جمع آوری اطلاعات از شبکه، نمایش مناسب آنها به کاربر و اعمال فرمانهای کنترلی لازم باشد. به منظور عدم ایجاد اختلال در سیستم توزیع قطعا دریافت اطلاعات و ارسال فرامین به رژنگتورها و ... باید به فرم صحیح انجام شود که لزوم تصدیق داده ها مستقل از کانال را نمایان می کند. در این مقاله برای تصدیق داده های کنترلی و دریافت فرمانهای صحیح استفاده از یک تابع یک طرفه با ساختار پیچیده

۱- مقدمه

سیستم اتوماسیون توزیع، امکان نظارت و کنترل شبکه های توزیع را فراهم می آورد و شرکت های توزیع را قادر می سازد که بتوانند به نظارت از راه دور، هماهنگ نمودن و اعمال فرمان روی تجهیزات به صورت آنی و از فاصله دور قادر باشند [۲]. در این راستا، علاوه بر پست های فوق توزیع با کنترل دیجیتال می توان در بخش توزیع نیز پست های پاساژ را اتوماسیون نمود، تا به راحتی بتوان به اطلاعات و وضعیت های محلی یک ناحیه دست پیدا کرد. اما به جهت آن که یک مجموعه فرامین از محل نظارت به پست ارسال می گردد امکان دارد که فرامین بالعکس ارسال یا دریافت شود و سبب قطعی و یا وصل بی موقع برق شود که زیانهای دو چندان را در پی خواهد داشت. لذا تصدیق اطلاعات مزبور دارای اهمیت فوق العاده می باشد. در مرجع [۱] برای تصدیق اطلاعات یک روش طرح امضاء مبتنی بر خم بیضوی ارائه شده بود که از امنیت بسیار بالایی برخوردار بود اما مشکل کندی سرعت آن الگوریتم، زمان تاخیری هر چند ناچیز را در تصدیق اطلاعات در بر داشت. در این مقاله استفاده از تابع درهم MD5 [۳] پیشنهاد شده است، اما با توجه به پارامترهای مبتنی بر تکرار توابع ساده غیر خطی پیشنهاد شده است. در این نوع توابع، تابع درهم MD5 انتخاب شده است. در پیاده سازی نرم افزاری این تابع انتخاب مناسب پارامترهایی مانند اشغال حداقل حجم حافظه ، عدم تعارن توابع اصلی بدنه سبب شده است که به سرعت و امنیت مناسب برای تصدیق اطلاعات دست پیدا کنیم.

تصدیق اطلاعات به روش مذکور از حیث سرعت نسبت به روش بیان شده در مرجع [۱] - که مبتنی بر امضاء دیجیتال بر پایه خم بیضوی می‌باشد برتری دارد اما از حیث میزان امنیت و شکسته شدن سیستم روش مرجع [۱] ارجحیت دارد هر چند که امکان وقوع تصادم برای تابع درهم پیاده سازی شده به سادگی میسر نمی باشد

این تابع سعی بر آن بوده که در پیاده سازی به سرعت بالاتر و امنیت بهتر دست یابیم که در نتایج حاصل از پیاده سازی مشهود می باشد.

نحوه تنظیم مطالب به صورت زیر است: در فصل دوم، تایید اعتبار پیام در سیستم اتوماسیون توزیع مورد بررسی قرار گرفته است. در فصل سوم، اصول و ویژگی های توابع درهم جهت تایید اعتبار بیان می شود. فصل چهارم، اختصاص به پیاده سازی تابع درهم MD5 دارد. در فصل پنجم، امنیت تابع درهم MD5 آمده است. در فصل ششم، مقایسه ای بین امضاء دیجیتالی بر پایه خم بیضوی با تابع درهم MD5 انجام شده است. فصل هفتم، نتایج بحث را در بر دارد.

۲- تائید اعتبار پیام در سیستم اتوماسیون توزیع

از دیدگاه IEEE سیستم اتوماسیون توزیع عبارت است از سیستمی که یک شرکت توزیع را به نظارت از راه دور، هماهنگ نمودن و اعمال فرمان روی تجهیزات توزیع در زمان حقیقی در مسافتهای دور قادر می سازد. در این راستا باید از بسترهای مخابراتی استفاده نمود.

استفاده از سیستم اتوماسیون برای شرکتهای توزیع مزایای زیادی منجمله کاهش هزینه های سرسام آور حفاظت و نگهداری شبکه، دسترسی دائمی به حجم زیادی از اطلاعات شبکه و تهیه نمودارها و گزارشهای آماری را در پی خواهد داشت. اما بدیهی است صحت اطلاعات دریافتی در هر کاری بسیار مهم می باشد، چه بسا که با ارائه اطلاعات غلط، بسیاری از عملکردهای لحظه ای و برنامه های آتی تحت الشعاع قرار بگیرد. عملکرد لحظه ای می تواند قطع و وصل ناصحیح دژنگتور یا سکسیونر باشد و عملکرد آینده را می توان برنامه ریزی (رشد بار و میزان مصرف سالیانه و... .. در نظر گرفت که هر کدام در جای خود خسارات فراوانی را به بار می آورد.

۳- توابع درهم

برای تائید اعتبار پیام می توان از توابعی موسوم به توابع درهم [۴] استفاده نمود. یک تابع درهم عبارتست از تابعی به صورت H=h(M) که دارای سه ویژگی زیر باشد:

- به دست آوردن مقدار تابع از پیام ورودی آسان است.
- بدست آوردن پیام ورودی از روی مقدار تابع مشکل است.
- با داشتن یک پیام M ، به دست آوردن پیام دیگر M' که در شرط $H(M) = H(M')$ صدق کند، مشکل است.

که در تعریف مذکور، مقصود از آسان و مشکل، پیچیدگی محاسباتی است.

معمولا مقدار خروجی تابع درهم بسیار کوچکتر از پیام ورودی می باشد که سبب راحتی ارسال آن از طریق کانال مخابراتی می شود.

برای تولید یک تابع درهم، ابتدا یک تابع ریاضی منطقی در نظر گرفته می شود، آنگاه با در نظر گرفتن یک مقدار اولیه، الگوریتم بازگشتی $h_i = f(M_i, h_{i-1})$ تا حصول به مقدار نهایی انجام می شود.

توابع درهم با توجه به بدنه تابع به کار رفته تقسیم بندی می شوند که از این نقطه نظر میتوان آنها را به دو گروه متداول مبتنی بر سیستمهای رمزنگاری قالبی و مبتنی بر تکرار توابع ساده غیر خطی تقسیم بندی نمود. در این مقاله تابع درهم MD5 که ازتکرار یکسری توابع ساده غیر خطی حادث می گردد، پیاده سازی می شود.

۴-پیاده سازی تابع درهم MD5

تابع درهم MD5 از پیام ورودی ۵۱۲ بیتی یک مقدار درهم ۱۲۸ بیتی تولید می کند. این تابع دارای یک حلقه اصلی ۴ مرحله ای است. الگوریتم تابع درهم به صورت زیر است:

۱- متن ورودی در بلوک های ۵۱۲ بیت پردازش می شود، که هر بلوک به ۱۶ زیر بلوک ۳۲ بیتی تقسیم می شود.

۲- خروجی ۱۲۸ بیت می باشد که از کنار هم قرار گرفتن ۴ بلوک ۳۲ بیت ناشی می شود.

۳- ابتدا پیام ورودی لایه گذاری می شود به گونه ای که طول آن ۶۴ بیت کمتر از مضرب صحیحی از ۵۱۲ گردد.

۴- برای لایه گذاری ابتدا یک بیت ۱ و سپس به اندازه لازم بیت ۰ به آخر پیام اضافه می شود.

۵- ۶۴ بیت که نشان دهنده طول پیام قبل از لایه گذاری است به نتیجه مرحله قبل افزوده می شود.

۶- چهار متغیر ۳۲ بیتی مقدار دهی اولیه می شوند:

A = ۰ x ۰۱۲۳۴۵۶۷

B = ۰ x ۸۹ abcdef

C = ۰ x fedcba ۹۸

D = ۰ x ۷۶۵۴۳۲۱۰

۷- برای هر کدام از بلوک های ۵۱۲ بیتی پیام ورودی حلقه اصلی الگوریتم تکرار می شود. حلقه اصلی چهار مرحله خیلی شبیه به هم دارد.

۸- مقادیر متغیرهای A و B و C و D در چهار متغیر a و b و c و d کپی می شوند.

۹- هر عمل، یک تابع غیر خطی روی ۳ تا از ۴ متغیر a،b ، c و d اعمال می کند.

$$F(x,y,z) = (x \wedge y) \vee (\sim x \wedge z)$$

$$H(x,y,z) = x \oplus y \oplus z$$

$$G(x,y,z) = (x \wedge y) \vee (y \wedge z)$$

$$I(x,y,z) = y \oplus (x \vee \sim z)$$

۱۰- نتیجه با یک زیر بلوک و یک مقدار ثابت جمع می شود.

۱۱- به اندازه S بیت شیفت گردشی به چپ انجام می شود (مقدار S متغیر است) و در نهایت آن را با یکی از چهار متغیر a, b, c و d جمع می کند و نتیجه را در یکی از آنها قرار می دهد.

۱۲- مقادیر c, b, a و d به C, B, A و D افزوده می شود وسپس الگوریتم برای بلوک بعدی پیام تکرار می شود.

اگر به طور مثال F(b,c,d) را در نظر بگیریم a با F(b,c,d) جمع می شود. حاصل با M_J و t_I جمع می شود که M_J ، J امین زیر بلوک پیام و t_I یک عدد ثابت است که در مرحله I برابر با جزء صحیح 2^{32} abs (sin I) است که I بر حسب رادیان است حاصل S بیت به چپ شیفت پیدا کرده با b جمع می گردد و در a قرار می گیرد.

FF(a,b,c,d,M_J,S,t_I) denotes a = b+ ((a + F (b,c,d) + M_J+ t_I)$^{<<<S}$)

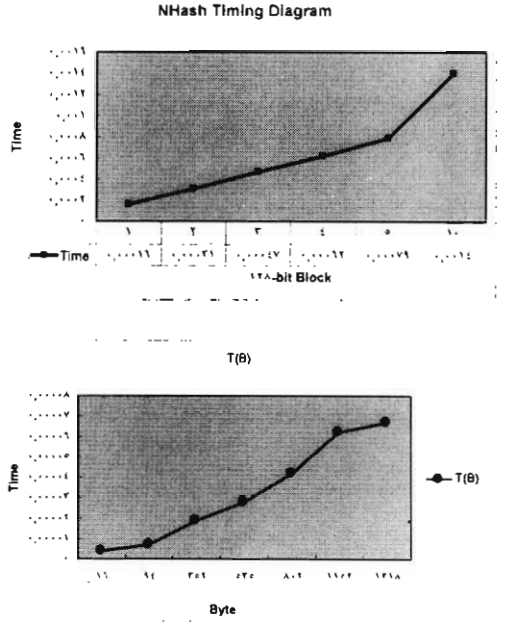
GG(a,b,c,d,M_J,S,t_I) denotes a = b+ ((a + G (b,c,d) + M_J+ t_I)$^{<<<S}$)

HH(a,b,c,d,M_J,S,t_I) denotes a = b+ ((a + H (b,c,d) + M_J+ t_I)$^{<<<S}$)

II(a,b,c,d,M_J,S,t_I) denotes a = b+ ((a + I (b,c,d) + M_J+ t_I)$^{<<<S}$)

در هنگام پیاده سازی، به کارگیری مناسب توابع F و G و H و I بسیار مهم بوده و در سرعت و امنیت الگوریتم تاثیر به سزا دارد.

در هنگام اجراء انواع مختلفی از توابع مذکور مورد آزمایش قرار گرفت که برای هر یک زمانهای متفاوتی به دست آمد که البته به یکدیگر از لحاظ کلی نزدیک بودند. (ضمیمه) نمودار زمان – بایت برای ورودی های مختلف تابع MD5 ذیلا رسم شده است.



(MD5 timing diagram)

با مطالعه این نمودار دریافت می شود که:

۱-به طور کلی MD5 از سرعت مناسبی برخوردار می‌باشد که در مقایسه با MD4 و N-Hash بسیار سریع تر می‌باشد.

۲-با افزایش طول پیام ورودی به طور متناسبی و با یک نمودار صعودی زمان اجرای الگوریتم افزایش می یابد که این موضوع حکایت از آن دارد که برای پیام های با طول بسیار بزرگ MD5 کارآیی خود را از دست میدهد که البته این موضوع با توجه به نوع فرمانهای ارسالی در سیستم اتوماسیون توزیع مرتفع می شود و ایرادی را بر MD5 وارد نمی کند.

۳-زمان مذکور با بهره گیری از توابعی با شرایط زیر حاصل می شود.

- ساختار کلی چهار تابع با یکدیگر متفاوت باشد.
- در هر تابع، تقارن وجود نداشته باشد.
- الگوریتم مناسب برای لایه گذاری

بالطبع پارامترهای مذکور، ضمن ارائه سرعت مناسب برای تابع MD5 ویژگی امنیتی لازم را نیز فراهم می آورد که ذیلا بحث شده است.

۵-امنیت تابع درهم MD5

در مجموعه توابع درهم مبتنی بر تکرار توابع ساده غیر خطی، توابع N-Hash و MD4 از اولین توابع پیشنهادی بودند که در مقابل حمله تصادم مقاوم نمی باشند. در سال ۱۹۹۵ دوبرتین در مقاله ای تحلیلی نشان داد که تابع درهم MD4 در مقابل حمله تصادم مقاوم نمی باشد [۵]. حمله سریعتری روی تابع مذکور در سال ۱۹۹۷ توسط کاسلمن انجام گردید [۶] . حمله ارائه شده توسط دوبرتین به دو بخش زیر تقسیم می گردد:

- تصادم داخلی
- مبتنی بر حمله تفاضلی و انتخاب مقادیر اولیه مناسب

حمله تفاضلی زمانی می تواند موفقیت آمیز باشد که تعدادی تصادم داخلی به وقوع بپیوندد. لذا قدم اول برای حمله به تابع مذکور یافتن تصادم داخلی است. در این راه، انتخاب مناسب مقادیر اولیه گام مهمی در کاهش تعداد معادلات منطقی موجود برای یافتن تصادم است.

اما در ارتباط باامنیت MD5 شرایط زیر وجود دارد:

اولا: یک مرحله به تعداد مراحل MD4 اضافه شده است پیچیدگی نسبت به MD4 بالاتر است.

ثانیا: به دلیل استفاده از توابع غیر متقارن در الگوریتم اصلی و توابع بهینه تری که در این مقاله استفاده نمودیم رخداد تصادم مشکل تر می شود.

لذا می توان گفت تابع درهم MD5 دارای ویژگیهای امنیتی مناسبی می باشد و قابل استفاده در کلیه مقاصد معمول تایید اعتبار می باشد.

۶-مقایسه تابع درهم MD5 با امضاء دیجیتالی بر پایه خم بیضوی

در مرجع (۱) برای تایید اعتبار سیگنالهای ارسالی و دریافتی در سیستم اتوماسیون توزیع یک طرح نو برای امضاء دیجیتالی بر پایه خم بیضوی ارائه شده بود که با توجه به آن که خم بیضوی در کلاس مسایل Np-Complete می‌باشد بالطبع دارای امنیت قابل قبول می باشد اما به لحاظ آن که خم بیضوی ذاتا دارای الگوریتم کند می باشد لذا سرعت تصدیق اطلاعات در سیستم اتوماسیون توزیع آنچنان که انتظار می رود بالا نیست. با پیشنهاد این مقاله مبنی بر استفاده از تابع درهم MD5 مشکل سرعت مرتفع می شود.

ضمنا با توجه به آن که تابع درهم MD5 در مقابل تصادم نیز تا حدود بسیار زیادی مقاوم بوده ودر زمانهای کوتاه به صورت آماری رسیدن به تصادم به سادگی میسر نمی باشد، استفاده از تابع درهم MD5 پیاده سازی شده برای مقصود تائید اعتبار سیگنالهای دریافتی در سیستم اتوماسیون از روش خم بیضوی به مراتب کارآتر است.

۷-نتیجه گیری

در این مقاله پس از بیان لزوم تصدیق اطلاعات در سیستم اتوماسیون توزیع، استفاده از تابع درهم MD5 بهینه شده پیشنهاد گردید. تابع مذکور با ایجاد تغییراتی نسبت به الگوریتم اصلی با نرم افزار Visual C++ پیاده سازی و اجرا گردید. با دستیابی به زمان های مناسب و با توجه به دارا بودن پیچیدگی های مناسب مشکل موجود در ارتباط با کندی الگوریتم پیشنهادی در مرجع (۱) مرتفع گردید و امکان تصدیق اطلاعات در سیستم اتوماسیون با سرعت مناسب و با یک الگوریتم امن میسر شد.

مراجع

[1] Faramarz faghihi, morteza esmaeili and morvarid sehatkar, "Information Security in Distribution Automation system by Elliptic curve", 17th International power system conference, 28-30 oct. 2002, Tehran – Iran – proceeding (4), control & protection _TeleCommunication – IT – pp. 89 – 97.

[2] Anil pahwa, "Flexible control of Distribution system", kansas state university, 1999.

[3] Alfred J. Menezes, paul cvan, Oorschot scott and A. vanstone, "Hand book of applied cryptography".

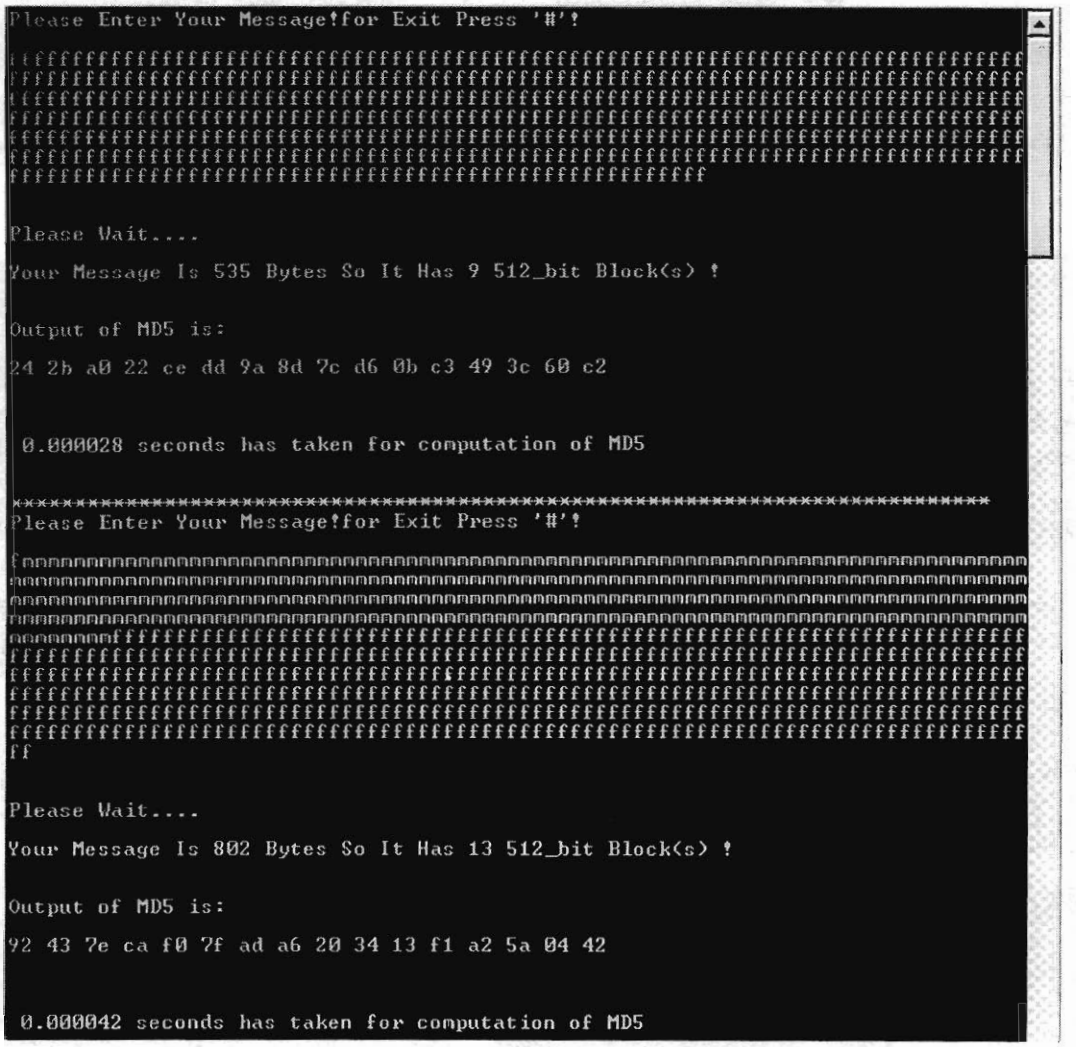
[4] Douglas stinson, "cryptography Theory and practice"

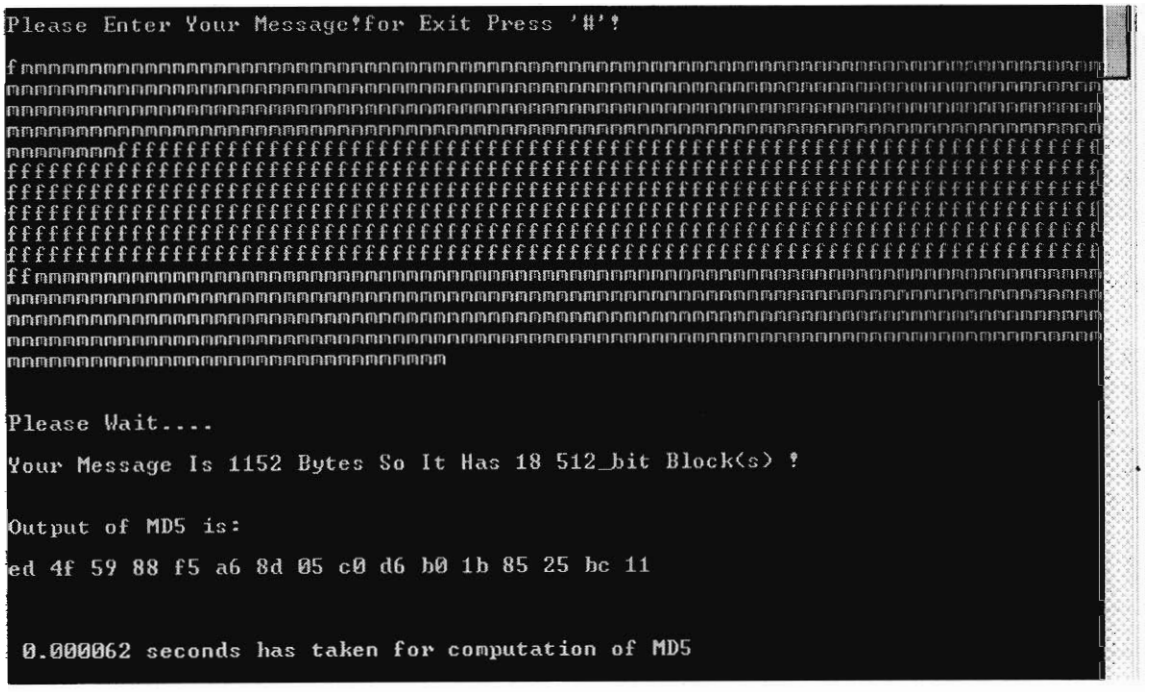
[5] H. Dobbertin, "Cryptanalysis of MD4" , FSE 96, pp. 53 – 69, 1996.

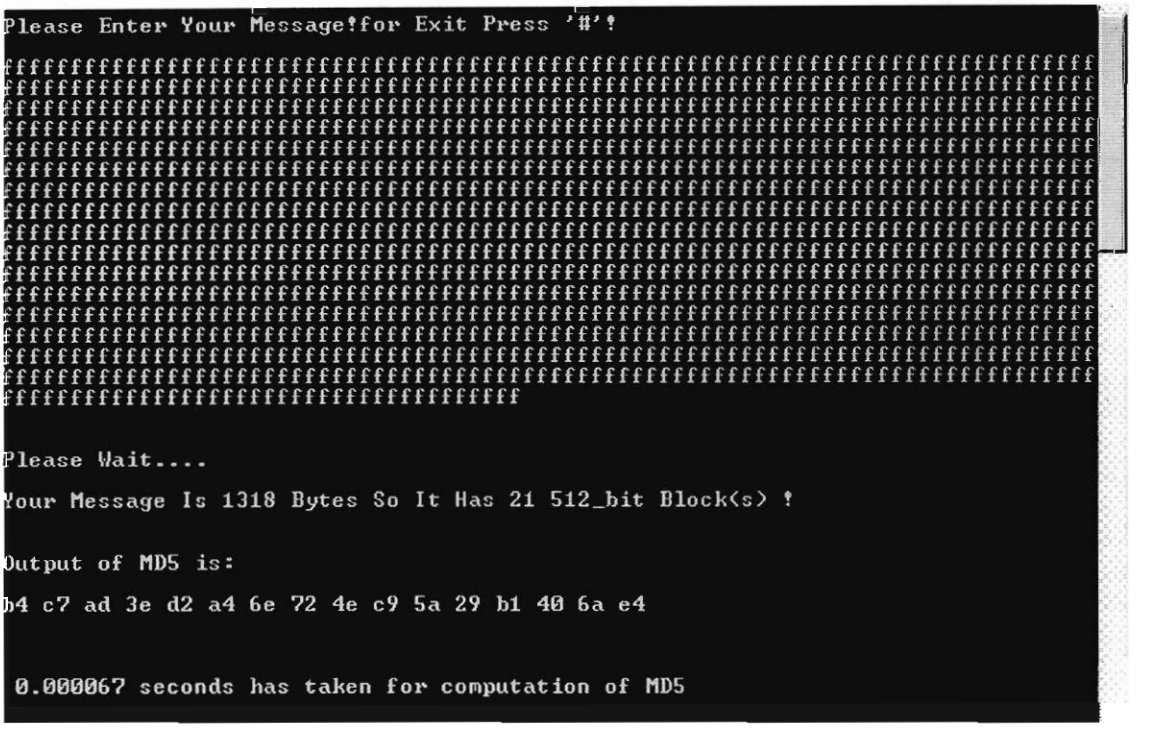
[6] P.R. Kasselman, "A Fast Attack on the MD4 Hash Function", information Theory, IEEE, 1997.

```
Please Enter Your Message!for Exit Press '#'!
salaam Faramarz!

Please Wait....
Your Message Is 16 Bytes So It Has 1 512_bit Block(s) !

Output of MD5 is:
64 28 5a 17 b8 5d 9e 76 64 c2 cf ba b8 e1 2c fa


 0.000004 seconds has taken for computation of MD5
```

```
Please Enter Your Message!for Exit Press '#'!
ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
ffffffffffffff


Please Wait....
Your Message Is 94 Bytes So It Has 2 512_bit Block(s) !

Output of MD5 is:
85 5c 7b 8c a1 1c d0 55 02 c3 24 e3 ea 6b 49 44


 0.000007 seconds has taken for computation of MD5

*****************************************************************************
Please Enter Your Message!for Exit Press '#'!
ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
ffffffffffffffffffffffffffffffff


Please Wait....
Your Message Is 352 Bytes So It Has 6 512_bit Block(s) !

Output of MD5 is:
31 98 b3 12 2c c0 6c bf cb 11 0b e4 b9 8e 12 f9


 0.000019 seconds has taken for computation of MD5
```


Please Enter Your Message!for Exit Press '#'!
Please Wait....
Your Message Is 535 Bytes So It Has 9 512_bit Block(s) !
Output of MD5 is:
24 2b a0 22 ce dd 9a 8d 7c d6 0b c3 49 3c 60 c2
0.000028 seconds has taken for computation of MD5
Please Enter Your Message!for Exit Press '#'!
Please Wait....
Your Message Is 802 Bytes So It Has 13 512_bit Block(s) !
Output of MD5 is:
92 43 7e ca f0 7f ad a6 20 34 13 f1 a2 5a 04 42
0.000042 seconds has taken for computation of MD5


Please Enter Your Message?for Exit Press '#'?
Please Wait....
Your Message Is 1152 Bytes So It Has 18 512_bit Block(s) ?
Output of MD5 is:
ed 4f 59 88 f5 a6 8d 05 c0 d6 b0 1b 85 25 bc 11
0.000062 seconds has taken for computation of MD5


Please Enter Your Message?for Exit Press '#'?
Please Wait....
Your Message Is 1318 Bytes So It Has 21 512_bit Block(s) ?
Output of MD5 is:
b4 c7 ad 3e d2 a4 6e 72 4e c9 5a 29 b1 40 6a e4
0.000067 seconds has taken for computation of MD5