پروتکل انتقال امن پیام در شبکه های مبتنی بر پردازش ابری

مسعود صدری، دانشگاه نبی اکرم (ص) – دانشکده فنی مهندسی – masood222@gmail.com
محمدعلی شریفلو، دانشگاه نبی اکرم (ص) – دانشکده فنی مهندسی – m.shariffloo@gmail.com
احمد کاظمی ، دانشگاه نبی اکرم (ص) – دانشکده فنی مهندسی – ahmad.kazemi20@gmail.com
تیم امنیتی AES

چکیده: این مقاله، شرح مختصری بر طراحی یک سیستم امن پیام رسانی تحت شبکه داخلی، طبق مدل پردازش ابری می باشد. برای حفظ امنیت پیام ها از سیستم رمزگذاری متن، بر اساس استاندارد AES و الگوریتم رایندل استفاده شده است. قطعه کدهای ذکر شده، پیاده سازی نرم افزاری این طرح بوسیله زبان برنامه نویسی C می باشد. به دلیل دید امنیتی موضوع، تمرکز اصلی بر روی رمزگذاری متون و ایجاد کانال های امن ارتباطی خواهد بود.
کلمات کلیدی: انتقال پیام – امنیت – پردازش ابری – رمزنگاری

۱- مقدمه

با ورود کامپیوتر به زندگی انسانها و فرا گیر شدن آن، همواره بحث امنیت در میان بوده است. بحثی که در آن از عدم حضور شخص سومی در مابین دو نفر که در حال گفتگو یا تبادل اطلاعات هستند، صحبت میشود. در این میان راههای گوناگونی برای حفظ امنیت تعبیه شده است که یکی از این راهها وجود الگوریتم مناسب برای رمز کردن متن گفتگو است، متنی که دو نفر برای یکدیگر ارسال میکنند. حال هر الگوریتمی خود نیز امن نیست و باید بهترین و استانداردترین الگوریتم را انتخاب نمود. انتخاب کلید مناسب با متن نیز بسیار تاثیر گذار است. در این مقاله سعی شده ساختمان یک نرم افزار پیام رسان از لحاظ امنیتی مورد بحث و تحلیل قرار گیرد و ایرادات وارده بر آن با ارائه راهکار بر طرف شود. همین طور، بحث پردازش ابری نیز به میان آمده است تا طرفین این نرم افزار بصورت یک ابر در نظر گرفته شوند. این موضوع میتواند در گسترش برنامه نقش مهمی را ایفا کند.

۲- شرح کلی نرم افزار

۲-۱ بررسی نحوه تعامل با سیستم عامل

اصل وجودی سیستم های عامل، جهت دسترسی راحت تر و سریع تر یک کاربر به سخت افزار است. به بیان دیگر، سیستم عامل سامانه ای را پدید می آورد که نرم افزارهای کاربردی بوسیله آن اجرا می شوند. نرم افزارهای کاربردی معمولا یا از طریق API ها و یا فراخوانی های سیستمی، به این سامانه دسترسی پیدا می کنند.

بعضی نرم افزارها به دلیل کد شدن و برنامه نویسی با زبان های خاص که اکثرا از توابع آماده و نصب شده بهره می برند (مانند محصولات مایکروسافت) حتما باید از یک به اصطلاح Platform خاص و یا همان سیستم عامل خاص (مانند ویندوز) استفاده کنند. بعنوان مثال اغلب برنامه هایی که تحت زبان C#.NET طراحی و برنامه سازی می شوند، به علت استفاده از توابع آماده و فراهم شده در Microsoft .NET Framework تنها در نسخه های سیستم عامل ویندوز قابلیت نصب و اجرا دارند[1].

طرحی که ما در نظر داریم در راستای پوشش انواع معایب امنیتی و مدیریتی می باشد؛ به همین منظور این نرم افزار را با یکی از زبان های برنامه نویسی C و یا Java پیاده سازی خواهیم کرد. زیرا جدا از قدرتمند بودن این دو زبان، ما در یک Platform خاص محدود نخواهیم بود و این برنامه را در سیستم عامل های دیگر مانند Linux به اجرا خواهیم در آورد.

البته در نسخه های بعدی مدل تحت وب را با زبان PHP ارائه می کنیم، تا تنها با استفاده از یک مرورگر و اتصال به سرور مرکزی از این سیستم بتوان استفاده کرد.

۲-۲ اجرا تحت شبکه داخلی

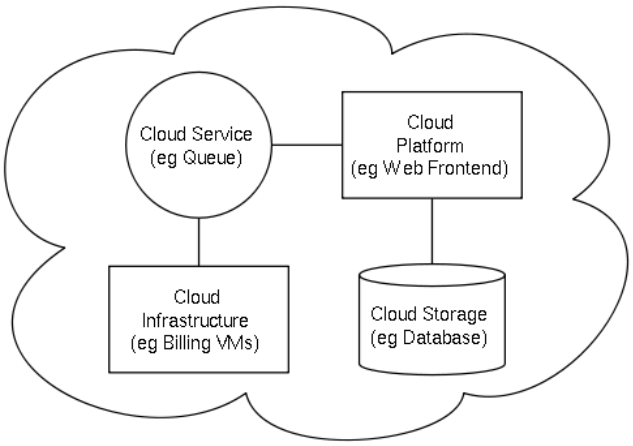
این طرح، بیان کننده ی یک سیستم پیام رسانی امن، تحت شبکه داخلی می باشد. اگر بخواهیم یک شبکه را بررسی کنیم چنین خواهد بود: بعنوان مثال ۶ سیستم کامپیوتری در داخل ساختمان وجود دارند که از طریق تکنولوژی LAN و بوسیله مسیریاب ها و مودم های بی سیم و یا باسیم به یکدیگر متصلند. بعد از نصب شدن برنامه بر روی Client ها، یک سری تنظیمات و سطوح دسترسی توسط Admin اعمال می شود. نکته مهم اینجاست که این سیستم از طریق یک سرور پشتیبانی می شود که این سرور توسط توابع تعیین شده، به صورت تصادفی (Random) از مابین Client انتخاب می شود. این موضوع تصادفی بودن سرور، یکی از سیاست های امنیتی در نظر گرفته شده است. وظیفه سرور این شبکه، دراصل پشتیبانی، کنترل کاربران و مسیرهای ارتباطی و همین طور حفاظت از آرشیو پیام ها می باشد.

سیستم اینگونه طراحی شده است که اگر به هر دلیلی (نفوذ، ناتوانی سخت افزارها، قطع شدن مسیر ارتباطی و ...) سرور دیگر نتواند پاسخگوی نیازها باشد، طبق الگوریتمی، مسئولیت پشتیبانی بر عهده ی یکی دیگر از Client ها قرار خواهد گرفت.

۲-۳ انجام عملیات، بر طبق پردازش ابری

مدل پردازش ابری یا همان Cloud Computing یکی از تکنولوژی های به روزی می باشد که هنوز دوران طفولیت خود را سپری می کند. این تعریف توسط موسسه ملی فناوری و استانداردها (NIST) ذکر شده است:"پردازش ابری مدلی است برای فراهم کردن دسترسی آسان بر اساس تقاضای کاربر از طریق شبکه به مجموعه ای از منابع پردازشی قابل تغییر و پیکربندی (مثل: شبکه‌ها، سرورها، فضای ذخیره‌سازی، برنامه‌های کاربردی و سرویس‌ها) که این دسترسی بتواند با کمترین نیاز به مدیریت منابع و یا نیاز به دخالت مستقیم فراهم کننده سرویس به سرعت فراهم شده یا آزاد (رها) گردد."

اصل این سیستم مدلی همانند فلسفه یونیکس می باشد که در آن چند برنامه به طور کاربردی و مستقل در محدوده ی خود، آن وظیفه ای که برایشان تعیین شده انجام را می دهند و از طریق یکسری واسطه به هم متصل می شوند. به شکل کلی، این سیستم ها از چند لایه ابر تشکیل شده اند و نحوه ی تعامل کاربر با محیط از طریق محدوده های ابری شکل انجام می گیرد و به بیان دیگر، سرویس ها تحت این خصوصیت انتقال پیدا می کنند. شکل زیر ترسیمی از همین تعریف می باشد[2].



شکل۱: نمونه ای از یک دیاگرام تحت پردازش ابری

به طور مختصر تعدادی از ویژگی های مفید این مدل برای سیستم مطرح شده را بیان می کنیم:

- زمان: کاربر سیستم، تسلط قابل توجه و محسوسی بر روی زمان پردازش اطلاعات دارد و می تواند باعث کاهش و یا افزایش آن شود.
- عدم وابستگی به نرم افزار و سیستم عامل خاص: این طرح بوسیله پردازش ابری، قابلیت پیاده سازی بر روی سیستم های کامپیوتری متصل به شبکه اینترنت را دارد که در آن تنها با استفاده از یک مرورگر، می توان از برنامه استفاده کرد.
- به اشتراک گذاری منابع: در این سیستم به جهت سهولت کار، افزایش سرعت و امنیت بالاتر، می توان منابع تعیین شده را در بین کاربران به اشتراک گذاشت تا در صورت نیاز پردازش ها در نقطه ای متمرکز شوند.
- امنیت: در پردازش ابری، اکنون به جای اشیاء (Objects) ما با ابرها برخورد داریم. این ابرها در سیستم، ولی به صورت مستقل وظایف مربوط به خود را انجام می دهند. در این حالت، مدیریت امنیتی فرایندها دقیق تر خواهد بود زیرا ابری از اطلاعات را به شکل جداگانه می توان مورد بررسی و تست قرار داد.

مدل های پیاده سازی اینگونه پردازش به این شرح است: ابر عمومی، ابر گروهی، ابر آمیخته و ابر خصوصی. برای طراحی این سیستم از مدل ابر خصوصی استفاده کرده ایم.

از ابر خصوصی جهت پیاده سازی پردازش ابری برای یک ساختمان و یا یک شبکه داخلی استفاده می شود. عمده مسئله قابل توجه در ابر های خصوصی، قابلیت کنترل و نظارت بیشتر بر روی پردازش ها می باشد. این مدل، در سیستم های داخلی پیاده سازی می شود و ارتباطی با خارج از سیستم نخواهد داشت. بنابراین طرف مقابل ما، تنها شبکه داخلی ساختمانمان می باشد و نیازی با پیاده سازی استانداردهای امنیتی خارجی وجود ندارد. در صورت مدیریت دقیق و اصولی امنیتی پردازش ها، حفاظت از اطلاعات را به طوری محسوسی می توان ارتقا داد.

در این سیستم، هر Client را بعنوان یک ابر در نظر گرفته ایم. با توجه به تعاریفی که ذکر شد، این مطلب به این معنی می باشد که هر کدام از سیستم ها، پردازش های مجزا و موازی در راستای یکدیگر دارند و به این ترتیب کنترل فرآیند اجرایی، دقیق تر و امن تر خواهد بود. همین طور، می توان عملیات ها و استانداردهایی را برای سیستم هر کاربر تعریف و پیاده سازی کرد.

۲-۴ انتقال امن پیام و فایل

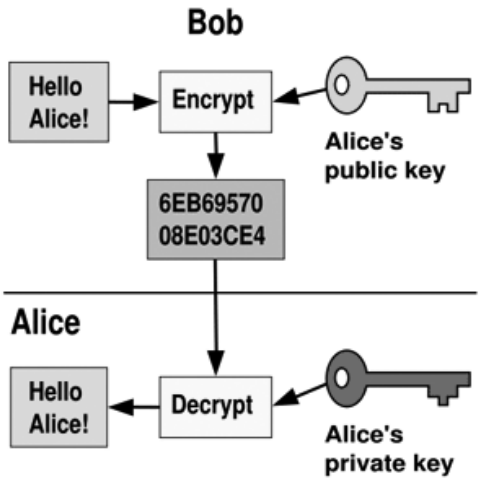
نرم افزارهای زیادی جهت انتقال پیام در سطح شبکه داخلی وجود دارند. از جمله می توان بسیاری از نرم افزاری های آماده بر روی سیستم عامل های Linux را مثال زد. اما نکته حائز اهمیت برای ما، امنیت محتوای این پیام ها و فایل ها می باشد. بعد از نصب این نرم افزار بر روی سیستم ها، توسط Admin سایر کاربرها تعریف شده و برای خود نام کاربری و رمز عبور در نظر می گیرند. شناسایی کاربران در سیستم و انجام پردازش ها، از طریق آدرس های IP مشخص شده می باشد. با اولین بار وارد شدن در نرم افزار و بعد از تعیین سطوح دسترسی، کاربر می تواند با کاربران حاظر به صورت دو نفری و یا گروهی ارتباط برقرار کند. در اغلب نرم افزارها، پیام ها ابتدا به سرور رفته و بعد به Client مورد نظر می رسند که در این حالت معمولا امکان حمله Man In The Middle زیاد است. سرور بعد از دریافت، بسته حاوی پیام را به کاربر مورد نظر ارسال می کند. البته برای حفظ امنیت در اینگونه سیستم ها، همیشه از رمزگذاری بر روی پیام ها استفاده می شود.

در این طرح، ما در نظر گرفته ایم که تمامی ارتباطات بدون هیچ واسطه و به شکل کاملا مستقیم صورت گیرد. اما تمام پیام ها در غالب Conversation History در پوشه ای در سرور ذخیره می شوند. به این منظور، دو Client مشخص، می توانند برای خود یک یا چند کانال ارتباطی امن برقرار کنند. این امر جهت کنترل دقیق تر بر روی مسیر ارتباطی، تعیین شده است.

از دیگر سیاست های امنیتی پیاده شده، مشخص کردن بایت های پوچ (Null)[3] می باشد تا از حملاتی که با استفاده از این خانه خالی بافر صورت می گیرد، جلوگیری شود. و همین طور، در بسته متنی برنامه، تنها کاراکترهای از قبل تعیین شده ارسال می شود؛ در این حالت از فرستادن دستورات سیستمی و Sell Code ها ممانعت خواهد شد. به بیان دیگر، در طول ارتباط بین دو یا چند کاربر سیستم، مسیری فراهم می شود که امنیت ارسال ها و دریافت ها را تا حدی تامین نماید.

۲-۵ رمزگذاری پیام ها

از مهم ترین نکات پیاده سازی شده در این سیستم، رمزگذاری بر روی پیام ها می باشد. هنگامی که کاربر پیام خود را وارد می کند، بعد از کلیک کردن بر روی گزینه ارسال، متن پیام بوسیله مقدار کلید عمومی از قبل تعیین شده و توسط الگوریتم خاص، رمز می شود و در کانال ارتباطی قرار می گیرد. بعد از دریافت توسط کاربر مورد نظر، بوسیله کلید خصوصی، پیام رمزگشایی می شود. همانطور که مشخص شد، برای این سیستم از رمزنگاری نوع کلید نا متقارن استفاده شده است. مقدار کلید توسط توابع سیستم تعیین می شود اما خود کاربر نیز می تواند تغییراتی را اعمال کند.



شکل۲: فرآیند رمزگذاری متن

برای الگوریتم رمز این سیستم، از استاندارد رمزنگاری AES و الگوریتم Rijndael استفاده شده است. به تعریف یک سری مفاهیم پیرامون این موضوع و نحوه ارتباط آن با سیستم مورد نظر می پردازیم.

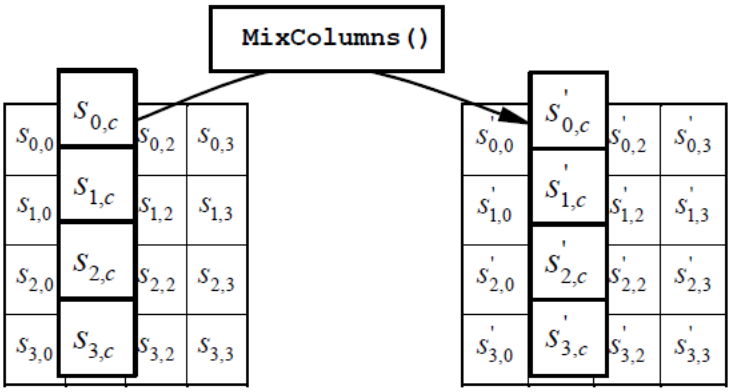
در این مدل رمزگذاری، رشته ای از متن با طول ثابت دریافت می شود. بعد از آن، عملیات اصلی نگاشت رمز بر روی متن اعمال شده و با همان طول اصلی به خروجی ارسال می شود. معمولا برای ارتقا امنیت از کلیدهایی با طول ۱۹۲ و یا ۲۵۶ بیت استفاده می شود. این الگوریتم شامل حدود ۱۶ مرحله می باشد؛ متنی که قرار است رمز شود، ابتدا در یک جایگشت خاص قرار می گیرد و سپس عملیات های وابسته به کلید بر روی آن اعمال شده و در نهایت در جایگشت نهایی قرار می گیرد. در اصل می توان گفت این دو جایگشت، عکس یکدیگر عمل خواهند کرد.

الگوریتم Rijndeal، یک سیستم رمز قطعه ای با طول قالب داده ۱۲۸، ۱۹۲ و 256 بیت است، طول کلید نیز مستقل از طول قالب ، ۱۲۸، ۱۹۲یا ۲۵۶ بیت می باشد. الگوریتم بسته به طول قالب داده و طول کلید شامل ۱۰ ، ۱۲ یا ۱۴ دور خواهد بود. این الگوریتم دارای ساختاری برای بسط کلید است که از روی کلید اصلی بسته به تعداد دورها، تعدادی زیر کلید تولید می‌کند که در هر دور به قالب داده اضافه می‌شوند. الگوریتم شامل سه تبدیل مهم MixColumn و ShiftRow وSubByte است که اولی

یک تابع جایگزینی غیر خطی و تامین کننده امنیت سیستم و دومی و سومی توابعی خطی برای افـزایش گسـترش و اخـتلاط الگوریتم اند. در این رمز قطعه‌ای، ساختار سیستم رمزگشا دقیقا مشابه سیستم رمزگذار نیست. هم چنین چون با افزایش طول کلید تعداد دورهای الگوریتم افزایش می‌یابد، زمان اجرا و سرعت الگوریتم به طول کلید وابسته است.[4] [5]

کمی به بررسی نوع عملکرد این توابع و پیادسازی نمونه ای از کد آنها می پردازیم.

MixColumn: این تابع روی آرایه حالت ستون به ستون عمل می‌کند. هر ستون به عنوان یک چنـد جملـه‌ای در میـدان دو بـه توان هشت در نظر گرفته می‌شود و در چند جمله‌ای ثابت a(x) ضرب می‌شود و به پیمانه $x4 + 1$ محاسبه می‌گردد.

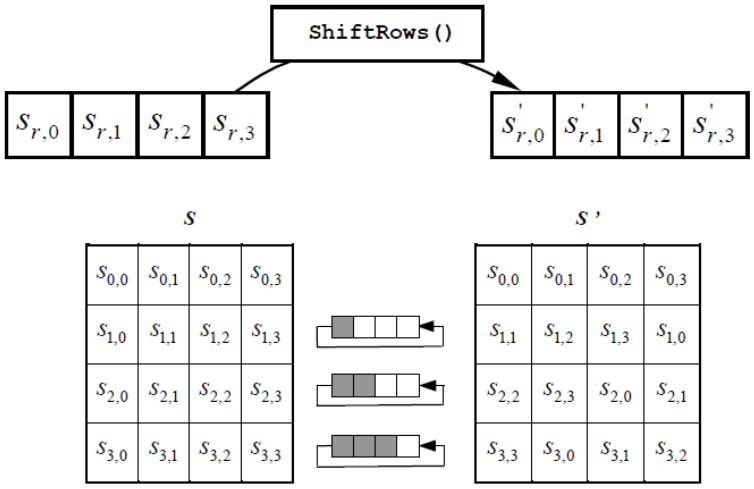


شکل۳: ماتریس تابع MixColumn

قطعه کد:

```
void MixColumns()
{
   int i;
   unsigned char Tmp,Tm,t;
   for(i=0;i<4;i++)
   {
      t=state[0][i];
      Tmp = state[0][i] ^ state[1][i] ^ state[2][i] ^ state[3][i] ;
      Tm = state[0][i] ^ state[1][i] ; Tm = xtime(Tm); state[0][i] ^= Tm ^ Tmp ;
      Tm = state[1][i] ^ state[2][i] ; Tm = xtime(Tm); state[1][i] ^= Tm ^ Tmp ;
      Tm = state[2][i] ^ state[3][i] ; Tm = xtime(Tm); state[2][i] ^= Tm ^ Tmp ;
      Tm = state[3][i] ^ t ; Tm = xtime(Tm); state[3][i] ^= Tm ^ Tmp ;
   }
}
```

ShiftRow: این تبدیل سه سطر آخر آرایه حالت را به تعداد معینی انتقال دورانی می‌دهد. برای اولین سطر ، r=0، انتقالی انجام نمی‌شود. تعداد انتقال دورانی در سه سطر آخر بستگی به Nb دارد به این ترتیب که برای Nb=8 انتقال های سه سطر آخر به ترتیب برابر ۱ ،۳ و ۴ و برای Nb<8 برار ۱،۲ و ۳ خواهد بود.



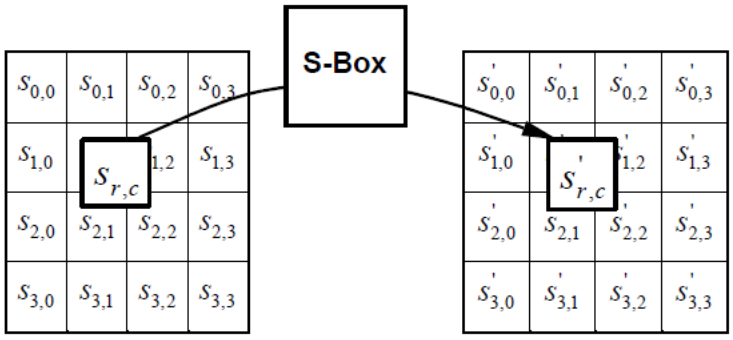
شکل۴: ماتریس تابع ShiftRows

قطعه کد:

```
void ShiftRows()
{
   unsigned char temp;
   temp=state[1][0];
   state[1][0]=state[1][1];
   state[1][1]=state[1][2];
   state[1][2]=state[1][3];
   state[1][3]=temp;
   temp=state[2][0];
   state[2][0]=state[2][2];
   state[2][2]=temp;

   temp=state[2][1];
   state[2][1]=state[2][3];
   state[2][3]=temp;
   temp=state[3][0];
   state[3][0]=state[3][3];
   state[3][3]=state[3][2];
   state[3][2]=state[3][1];
   state[3][1]=temp;
}
```

SubByte: این تابع یک تابع غیرخطی است که به طور مستقل روی بایت های آرایه حالت عمل کرده و به جای هر بایت به کمک جدول S-Box یک بایت جدید قرار می‌دهد. این تبدیل معکوس پذیر است و از دو تبدیل تشکیل شده: ابتدا معکوس ضربی بایت مورد نظر محاسبه می‌شود . معکوس " ۰۰ " را " ۰۰ " در نظر می‌گیریم . سپس تبدیل مستوی (Affine) روی بایت مورد نظر اعمال می‌شود.



شکل۵: ماتریس تابع S-Box

قطعه کد:

```
void SubBytes()
{
   int i,j;
   for(i=0;i<4;i++)
   {
      for(j=0;j<4;j++)
      {
         state[i][j] = getSBoxValue(state[i][j]);

      }
   }
}
```

۲-۶ سطوح دسترسی

در ابتدا ذکر شد که این سیستم بوسیله یک سرور که هویت نامشخصی دارد پشتیبانی می شود. از جمله وظایف این سرور، تعیین میزان حافظه و پردازش ها برای سایر کاربران و به یک اندازه می باشد. در این میان، کاربر اصلی یا همان Admin تعیین خواهد شد. مدیر سیستم، می تواند برای سایر کاربران سطوح مختلف دسترسی مشخص کند. تعیین طول مدت صحت عملکرد کاربر، ایجاد یک کاربر جدید، ارسال اخطار و یا پیام گروهی، بستن تمام پنجره های گفتگو و امکان تغییر رمز عبور کاربران را می توان بعنوان اختیارات مدیر سیستم مثال زد. یک کاربر عادی، قادر است کاربر و یا کاربران مورد نظر خود را آگاه سازد و با آنها شروع به گفتگو کند. همین طور می تواند فایل های مشخصی را ارسال کند (جهت ارتقا امنیت سیستم، تنها یک سری فایل با فرمت از قبل تعیین شده قابل تبادل می باشند).

۳- نتیجه گیری

موضوع امنیت، یکی از مباحثی است که اکثر اوقات کمتر مورد توجه کاربران و حتی مدیران قرار می گیرد. اما در اصل بعنوان مهم ترین مسئله می توان ذکر کرد. یک سیستم کامپیوتری، مخصوصا بعد از اتصال به یک شبکه، همیشه تحت تاثیر انواع تهدیدات امنیتی قرار می گیرد. معمولا کاربران با ذکر این جمله که ما شخص مهمی نیستیم! به حفاظت از حریم شخصی خود توجهی نمی کنند در حالی که اینجا دو نکته مطرح است: ۱- هر کاربری یقینا اطلاعات ارزشمندی دارد، همانند نام کاربری و رمز عبور ۲- از طریق در اختیار گرفتن کنترل کاربرانی با سطح دسترسی پایین تر، می توان به دسترسی بالاتر در شبکه رسید. پس امنیت در شبکه، یک مقوله ای نیست که تنها با دید شخصی بتوان به آن پرداخت.

سیستمی که در این مقاله مطرح شد، در راستای ارتقاء سیاست های امنیتی شبکه و همین طور جلب توجه کاربران به موضوع امنیت پیام هایی که ارسال و دریافت می کنند بود. این طرح قابلیت پیاده سازی به طور کامل و همین طور پشتیبانی، در شبکه داخلی را دارد. نرم افزارهایی برای این موضوع وجود دارند، اما به دلیل متن باز نبودن و همین طور بارگذاری از طریق شبکه اینترنت، نمی توان به امنیت آنها خیلی مطمئن بود. در حالی که این نرم افزار به شکل بومی طراحی شده است و به تمام مراکز و سازمانهایی که به محتوا و امنیت پیام ها توجه بیشتری دارند، توصیه می شود.

همیشه، نفر سوی وجود دارد، که شما از حضور آن مطلع نیستید. و همیشه، محتوای پیام شما مهم و حداقل برای خودتان ارزشمند می باشد. امیدواریم با پیاده سازی این مقاله توانسته باشیم این موضوع را برای شما روشن کنیم.

مراجع

[1] **Parallel Programming with Microsoft Visual Studio 2010 Step By Step**. Donis Marshall.
[2] **Cloud Security and Privacy**. Tim Mather, Subra Kumaraswamy, Shahed Latif.
[3] **CEH, Certified Ethical Hacker**. Kimberly Graves. **Hacking: The Art Of Exploitation**. Jon Erickson.
[4] **Applied Quantum Cryptography**. Christian Kollmitzer.
[5] **Hardware Implementation Of AES Algorithm**, Marko Mali, Franc Novak, Anton Biasizzo.