БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ • BULGARIAN ACADEMY OF SCIENCES

ПРОБЛЕМИ НА ТЕХНИЧЕСКАТА КИБЕРНЕТИКА И РОБОТИКАТА, **64**
PROBLEMS OF ENGINEERING CYBERNETICS AND ROBOTICS, **64**

София • 2011 • Sofia

Probable Risks Concerning Security in Email Communication and a Protection Approach

Nikolay Dokev, Ivan Blagoev

New Bulgarian University, 1618 Sofia
Emails: n.dokev@nbu.bg blagoev.i@gmail.com

Abstract: *This paper describes the risks to privacy when using communication by electronic mail. Different ways are discussed that compromise the email, as well as the approaches to avoid them and ensure the reliability of an email. A method is proposed and a software product developed that combines the advantages of the most commonly used technologies. The areas of its applications are discussed.*

Keywords: *Email communications, security, x509 certificates, S/MIME, email encryption.*

1. Introduction

The development of global information network and the necessity for communication among people has lead to the constant need of electronic mail (email) usage. Even users who do not often use Internet and whose business is not connected with this, make the most of the electronic mail as a communication tool. On the other hand, a majority of the users are completely dependent on this way of communicating. It is considered that the electronic mail is the basic reason for fax service fading. Internet has, of course, other ways of information exchange among users, but the electronic mail is still most widely spread.

However, can we be certain in email letters authenticity?

It is a fact that this technology is very popular because it is cheap and easily accessed by every user. Each company, no matter how small it is, could get its own email server. Nevertheless, the low price, the small maintenance expenses, the efficiency and the wide distribution, do not mean that this communication method does not bring a lot of risks. Further on we shall discuss some possible threats that are often neglected by the users, while operating this service in its common form.

2. Ways of email compromising

One of the simplest, but also mostly neglected possibility, is the user's carelessness. Very often at places, where different people have access to a given computer, the user might forget their computer unlocked and the profile logged in, while being away at this very moment. Thus any person, having access to the computer system, might take advantage and send a false letter on behalf of the email account owner. This probability is so actual and frequently met that even the security audits of some standards, like ISO, require obligatory locking of the computers during the time when the users are not in front of them. Some hardware manufacturers have even designed devices that "lock" the computer when the user walks away. In order to be sure in the emails received, some users trace their origin, checking their headers – which server the letter comes from and whether the IP address corresponds to the actual one. Anyway, for this type of a fraud, such checks are not always efficient.

Another way is when the hacker re-directs the connection of a user and mediates between the user (client) and the server. This means that the haker controls the communication with the client and responds in the way the server would when the client wants access. At the same time he is communicating with the server as if being the actual client. This gives a possibility to the hacker to modify the content coming from the client, to derive information from it and to send false answers, as if generated by the actual server. This type of hacker's attack is one of the most dangerous and called "man-in-the-middle". In this case the correspondence may be easily counterfeited. Unlike the previous fraud described, this attack requires considerable knowledge in information technologies aspect. The hacker has to be patient and stubborn in order to succeed. However, this fact must not make us calm and neglecting this thread. In addition, once the attack becomes successful, it remains invisible for the users affected, because in most of the cases the electronic messages pass through the computer networks entirely unprotected as a clear text and nothing counteract it.

The use of an "Open Relay" – unprotected mail server enables sending of a false letter on behalf of any of the users. The Internet hackers often use such servers to distribute spam or fraud that may be with a longer and more serious purpose. The fraud result is not always immediately revealed. In many cases the hacker is able to derive important information which is the clue for other greater aims he wishes to achieve. Thus, besides the correspondence, it might appear that the damages are much more serious (Fig. 1).

One of the approaches, protecting against this type of counterfeited mail, is to maintain and constantly update the so called “black lists” with unprotected functioning mail servers. Thus, if a mail server receives a letter and the IP address of the sender is in this “black list”, this letter will not be re-sent, but blocked. Another way is by reverse Domain Name Server (DNS). When the mail server receives a letter, the server checks whether the IP address of the sender corresponds to the domain, generating the letter. In case it does not correspond, this letter is blocked. Anyway, both ways have some shortcomings and do not guarantee complete protection. In the case of supporting “black lists” with IP addresses of mail servers, the delay from the detection up to publishing in the list an unprotected mail server, might be fatal and the false mail – already received by the “victim”. As for the reverse DNS, it is almost inapplicable, because it turns out that a large part of the mail servers in Internet, though real, do not possess correct corresponding reverse DNS.

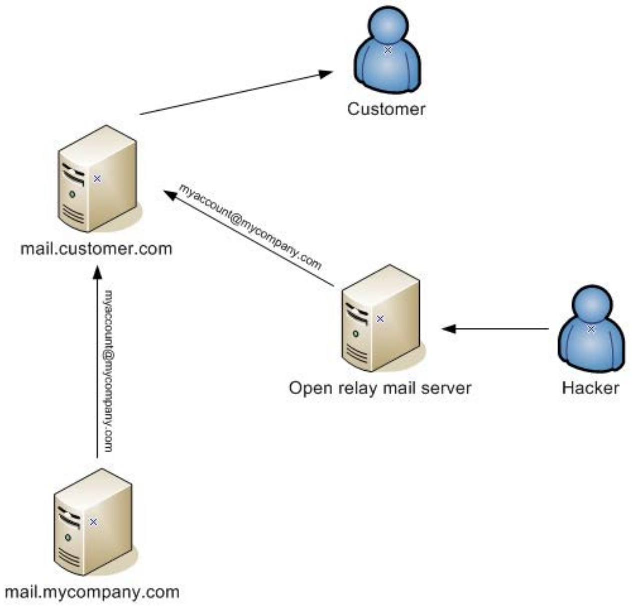


Fig. 1

The client programs for operation with electronic mail usually use POP3 protocol for letters reception. This protocol, as well as SMTP, is a protocol of high level, their commands and the information they transmit are in clear text. Their content could be read by “listening in on” the TCP/IP protocol in the global or local computer network. The user’s name and password that might become visible to the hacker through this attack, may serve for access to the mail box of their owner, as well as for the authentication required by the mail server, while delivering false letters. Fig. 2 shows an example of this type of attack.

In order to connect the example from Fig. 2 with practice and to give clearer explanations, we propose the following test experiment. For this purpose we have selected an arbitrary provider of free electronic mail in Internet that provides services which are not at all protected to his clients. We have registered a mail account with the name “userforest” and password “unsecure”.

The tool used in the experiment is a program for listening in on to the network traffic. No matter what operation system is used, there are always certain methods to trace the passing through or going into the computer TCP/IP traffic. In this example a 64-bits Windows operation system is used. Under Windows, as well as under other operation systems, there is a large variety of software, that may be used to “listen in on” TCP/IP traffic and a great part of these products possess quite flexible and developed interface for the aim set.

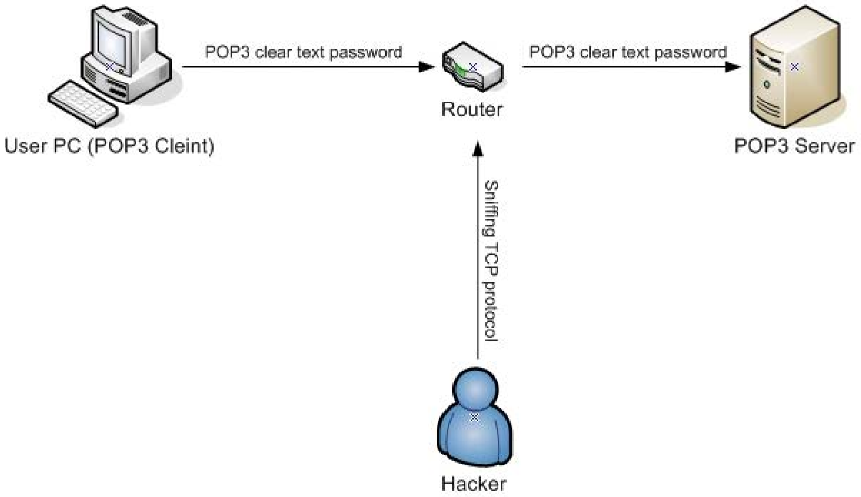
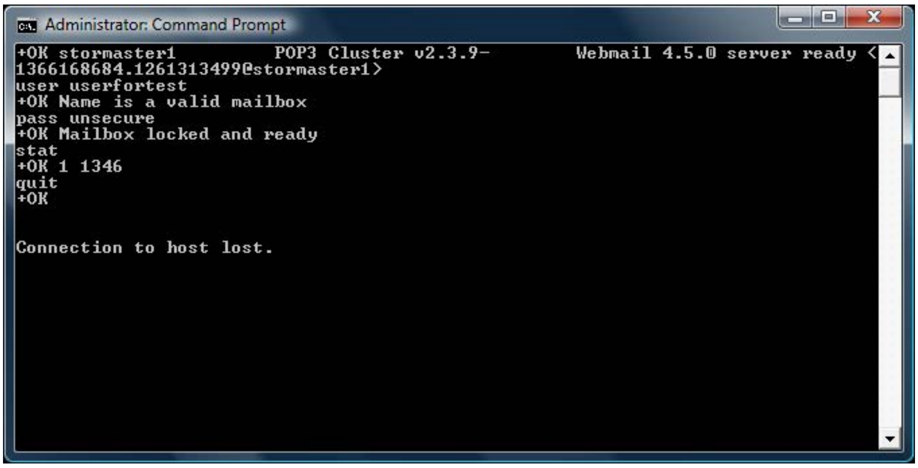


Fig. 2

Microsoft Network Monitor is used in the example, which may be free downloaded from the official Microsoft site. This is a very good product, designed with the purpose to aid software developers in detecting errors when creating network software, or in removing communication problems. After the installation of Microsoft Network Monitor, a network interface must be defined, and then traced. In this case the following filter is applied:

IPv4.Address ==XXX.XXX.XXX.XXX and IPv4.Tcp.Port ==110.

The IP address of the server, towards which the client will turn to (POP3 Mail Server IP Address) is placed at XXX.XXX.XXX.XXX. The port is 110, since this is the standard port for this service. In random cases it can also be changed by the administrator of the mail server considered. But this is not difficult to be discovered and not a reason to stop and protect the process of “listening in on” users’ names and passwords. The filter is necessary because a large quantity of packets pass through the network, so that if all are caught and stored, this would considerably hamper the processing and deriving of the information needed. After configuration, the “Apply” button is pressed, and then “Start”. Then the hacker has only to wait for the moment when the users check their mail. In order to show the example in details, we shall indicate what commands would send a mail client, using the built in Windows Telnet client. This makes possible to see the communication between the client and the server in POP3 protocol (Fig. 3).



```
+OK stormaster1          POP3 Cluster v2.3.9-        Webmail 4.5.0 server ready <
1366168684.1261313499@stormaster1>
user userfortest
+OK Name is a valid mailbox
pass unsecure
+OK Mailbox locked and ready
stat
+OK 1 1346
quit
+OK


Connection to host lost.
```

Fig. 3

After completing the communication, we shall check the content of the logs in an apriori prepared for the purpose Microsoft Network Monitor. The information stored in the log is more than the necessary amount, including a 16-bit presentation of the packets content, that is why we shall display just the relevant part of it:

POP3 response: +OK stormaster1 POP3 Cluster v2.3.9 Webmail 4.5.0 server ready <165147126.1261227502@stormaster1>
POP3 command: user userfortest
POP3 response: +OK Name is a valid mailbox
POP3 command: pass unsecure
POP3 response: +OK Mailbox locked and ready
POP3 command: stat
POP3 response: +OK 1 1346
POP3 command: quit
POP3 response: +OK

From this example it becomes clear that using the service in this form, besides the risk to reveal the user’s name and password, enables the hacker to read our whole correspondence, even without using our name and password. And all this is possible only by “listening in on” the router, where our traffic passes, because he can take the content of POP3 protocol exactly in the form, in which it is accepted and sent between the mail server and mail client. The fact that the users do not usually realize up to what extent they are unsecured and what risks their email is subjected to, is quite alarming. Thus, trusting a service which they do not know in details and accepting the idea that the password is familiar just to them, they can easily become fraud victims. An approach to protect against this way of “listening in on” attack is through SSL encrypting of the communication between the client and the mail server. Nevertheless, practice shows that most of the functioning mail servers are not defended in this way. Even if we assume that there are some exceptions, we cannot be certain that all our correspondents are protected against this or other types of attacks.

The usage of free and common Web mail services in Internet has also its risks. The convenience to have a suitable and similarly displayed mail client wherever we go is very attractive. But it brings the risk of account compromising by some programs, such as Key Logger. The use of different computer networks, that are not checked and not secure, exposes us to such threads. The programs of this type are "invisible" for the users, they are apriori introduced and like network traffic "listening in on", they also save information, but in this case the object is the user's keyboard. The entire input by the user is caught and stored in a secret log that is read by the hacker responsible for the attack.

Let us assume that the computers we use are not an object of hacker's counterfeit. The users quite often do not know the fact that not every server, suggesting free email in Web, is protected against "listening in on" the network traffic. HTTP is a text-based protocol of high level, which can also become vulnerable to "listening in on" (Fig. 4). When entering the user's name and password for access to Web mail box, it is possible to read them, as evident in the following example.

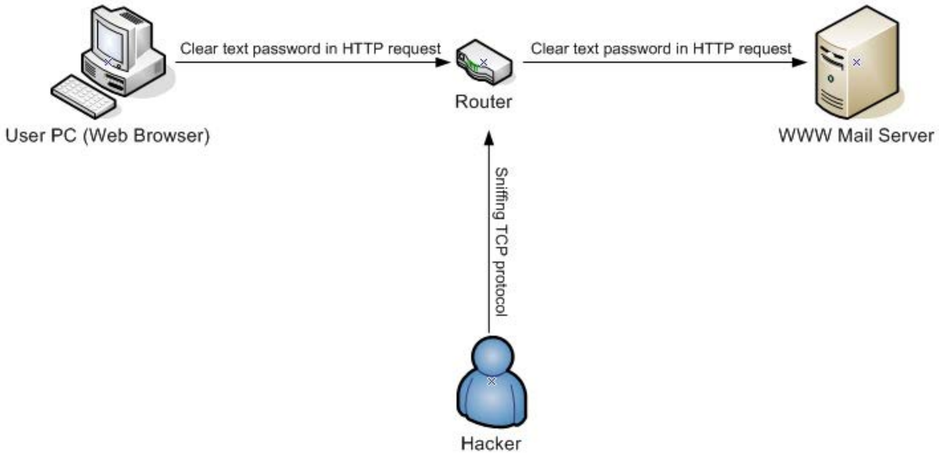

Clear text password in HTTP request
Clear text password in HTTP request
User PC (Web Browser)
Router
WWW Mail Server
Sniffing TCP protocol
Hacker

Fig. 4

For this example we register in an arbitrary and free Web mail, using the already mentioned name "userforest" and password "secure".

The transmission of the user's name and password is done by sending the format from the browser in a Web page for user's authentication. For the example we shall again use the already familiar Microsoft Network Monitor. The setup is similar to the previous example, with the only difference that for filtration will be used the packets, passing through port 80. It is necessary to limit the tracing only to packets going towards the IP of the server providing the email. This will decrease considerably the volume of the collected information and will make the log easier for investigation. After everything is prepared, we shall log in the Web mail, as the user (victim) would do in his Internet browser. The next step is to review the log, collected by Microsoft Network Monitor. Since the log from HTTP protocol is much larger than the log of POP3 protocol, we shall again show the shortened content only, which reveals the name and the password for access:

```
    POST http://httpmailservername/base/mail/redirect.php HTTP/1.0
    Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-
shockwave-flash, application/x-ms-application, application/x-ms-xbap,
application/vnd.ms-xpsdocument, application/xaml+xml, application/vnd.ms-excel,
application/msword, */*
    Referer: http://httpmailservername/
    Accept-Language: bg
    Content-Type: application/x-www-form-urlencoded
    Proxy-Connection: Keep-Alive
    User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1;
.NET CLR 2.0.50727; .NET CLR 1.1.4322; .NET CLR 3.0.4506.2152; .NET CLR
3.5.30729)
    Host: httpmailservername
    Content-Length: 179
    Pragma: no-cache
    Cookie: OAID=ab0effc4d04e3234e3ac6f43ad5e0d6a; t=m_a.-70_s.f;
__utma=189809463.1027347414.1260963270.1260964887.1260967227.4;
__utmb=189809463.1.10.1260967227;
__utmz=189809463.1260963270.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(no
ne);
OACBLOCK=2576.1260963236_204.1260963237_25.1260963237_2564.1260964
706_2560.1260964769_2534.1260965171_1733.1260965173_2276.1260965173_1
743.1260965173_2468.1260965173_2614.1260965174_350.1260965174_1866.126
0965177_2374.1260965178_2588.1260965181_1735.1260965181_2584.12609651
88_971.1260965188_1747.1260965267_2238.1260965272;
OACCAP=2576.2_204.13_25.29_2564.3_2560.8_2534.3_1733.2_2276.2_1743.2_
2468.1_2614.1_350.4_1866.4_2374.2_2588.1_1735.1_2584.1_971.1_1747.2_2238.
2; OABLOCK=5662.1260964706_5442.1260965173; OACAP=5662.3_5442.1;
session=480664e2edfffe90e08518e8fc0edcdb;
imp_key=480664e2edfffe90e08518e8fc0edcdb;
auth_key=480664e2edfffe90e08518e8fc0edcdb; _OACCAP[25]=1;
_OACCAP[204]=1; __utmc=189809463; _OACCAP[2560]=1

    session=480664e2edfffe90e08518e8fc0edcdb&actionID=&bui=&url=&load_f
rameset=1&autologin=0&ie_version=6%2C0%2C2900%2C5512&server_key=cyr
us&imapuser=userfortest@httpmailservername&pass=unsecure
```

The user’s name and password obtained from HTTP traffic are transmitted in the log as clear text. The values of the variables “imapuser” – for user’s name and “pass” – for the password are underlined herein.

After the experiments accomplished we could conclude that we cannot be entirely sure in the reliability of our email correspondence. In addition, even in case a person considers that their system is well defended, they cannot be sure in all electronic letters received. However, the information exchanged is not always so important and confidential and this is a prerequisite not to bother too much about the fact that a malicious user might falsify it. That is why most of the time we are using our email, we do not become victims of such a fraud. Our idea is to pay attention to the cases when we must be absolutely certain in the information received through email. And particularly in these cases it will be nice to have a solution, which will decrease to minimum the risk of counterfeits. It is also possible to integrate such a solution in an information system, re-sending information that is significant for the users. For example, storing contracts, bank accounts, credit card numbers, etc., that requires also notification about certain events.

3. Technologies of email protection

We propose and describe some technologies that may be combined and applied to be a solution of the problem.

S/MIME (Secure/Multipurpose Internet Mail Extensions) is a protocol developed by RSA Data Security Inc. to avoid counterfeiting of email messages. It is created on the basis of the existing MIME protocol that makes it easily integrated with the existing software email products. S/MIME is developed on the widely spread existing standard which has enabled its rapid deployment among different email clients and different operation systems. This, on its hand, made email sending and receiving among users with different operation systems possible. Thus a Windows user possessing Outlook Express may send an email to a Unix user, applying Netscape email client. At that, it is not necessary to install additional programs or software for this purpose.

S/MIME takes care to provide the following cryptographic services in email messages application: authentication message integrity and non-repudiation of origin (using digital signatures), and privacy and data security (using encryption).

X509 Certificate is a digital certificate of X509 type that can assist in protection, because: it connects with a pair of a private and a public key and an individual Distinguished Name and address of the email for owner’s identification. That is why in order to connect an encrypted and/or signed email with a certain digital certificate, it is necessary the email of its author to coincide with the published one in the digital certificate. X509 certificates are built as a hierarchical model, on its top a Root certificate is placed, and each lower one is signed by the above located. That is why the official providers of digital certificates publish them in a certificate chain, where the certificates are ranked according to their hierarchy. Other methods of protection concern the certificate expiry date. It is assumed that if a person attempts to discover the private key of a given certificate by guessing, the certificate expiry date will come. Another approach is CRL (Certificate Revocation List) – a list with certificates that are invalidated due to fears for loss or compromising. Last, but not least in importance, come the hardware crypto-processors, which must protect the private key of the certificate against malicious hackers' software, incorrect application or other compromising methods.

CSP (Cryptographic Service Provider) is a software library, which contains powerful implemented cryptographic algorithms. It is to be combined with a digital certificate, which possesses a private and a public key. An additional convenience is that it also contains implemented interfaces for control of the hardware cryptographic modules and crypto-processors (smart cards). This guarantees that the private key, belonging to the digital certificate is protected to a high security degree and any fraud with it becomes almost impossible.

The principle of **hardware protection with the help of a smart card** is that either it or a similar hardware tool contains the private key of the user. When there is a need to apply the private key to any cryptographic operation, it does not leave the device and cannot be read by a malicious user through the computer system. CSP contains interfaces which allow the encrypting, sending and processing of the data from the crypto-processor on the smart card, but not in the memory of the computer system. Thus the private key is applied but at the same time it remains protected. After the description of the operation principle here, there remains the uncertainty that it is possible to send any information by a malicious program that will use the program interfaces of the corresponding CSP. Thus data will be signed that will make a hole in all security precautions up to the moment. But in fact, when sending the data that must be encrypted with the help of the private key, which means also turning to the crypto-processor, the operation will be blocked until the operator enters his PIN code. Thus the access to the device is controlled and the system cares about notification of the user about the start of a process which wants to use the smart card, and after that the user decides whether the operation will continue.

When encrypting an electronic letter with the help of the technologies above mentioned, an encrypted electronic letter with S/MIME format is generated. The public key of every of the letter users is taken out of their certificates and they are identified by the name of the publisher of their digital certificate and serial number. When a message is delivered to a recipient, the electronic mail software detects the content of the message by its defined content type. After that the digital certificate, used to encrypt the content is applied, in order to decrypt it with the help of the private key. The security level in this method is quite high and could give us some certainty that our electronic mail is inaccessible. Anyway, security has different aspects. Let us assume that the digital certificate is lost or re-issued with a new pair of a private and a public key and it does not keep the previous pair of keys. This would cause impossibility to open and read your own correspondence, encrypted with the use of the previous pair of keys. Even the letter author, after encrypting it with the help of the encrypting algorithms, would not be able to access it. While reviewing the folder of sent items in your client's mail, if there are some encrypted ones, you will notice the letters as available in the list of sent items, but unable to be downloaded. This fact has another advantage as well. In case a malicious person reviews secretly the sent mail, he would not be able to access the letters encrypted. Moreover, wherever such a letter passes, through different routers and servers, it will be unreadable. And the recipient may hide its content again by excluding the cryptographic module or the SIM card with the digital certificate on their computer.

It is necessary to know that it is not obligatory to be an owner of a digital signature for encrypting and signing, and have obligatorily an additional external cryptographic device, as above described. You could have a completely functioning digital certificate, entirely stored in the operation system. But in this case the security degree would not be so high. The reason for this is that the private key, which is the most important security element, will be stored and accessible in the memory of the computer system at moments, when it must be applied. This particular moment causes the greatest thread and a risk exists that at this time it might be copied from memory. During the remaining time it will be encrypted and saved by the operation system.

However, we cannot mention the cases when there is no necessity for such serious precautions, because the information transferred does not require it, and we also consider our computer system sufficiently protected. Hence, the compromise of not using cryptographic hardware is excusable. Such a letter, similarly to the generated by the cryptographic hardware, will pass unread through many Internet machines, until reaching the recipient. That is why, the risk degree and the expenses for its avoiding must be determined according to our necessities.

A new approach is suggested, combining the technologies, above discussed, that avoids the risks described in the paper. This is achieved, taking advantage of the good aspects of each one of the technologies selected.

SMIME format is used to transmit an encrypted or signed letter by email. We encrypt the letter by a symmetric encrypting algorithm. But in order to be sure that the symmetric approach will not be compromised, we shall combine it with the security of the digital certificate and its protected private key. Thus it is possible to apply an asymmetric algorithm to the key encoding that we have used in the symmetric algorithm. Adding a hardware crypto-processor, that we have chosen to defend the digital certificate and its private key, we shall obtain as a result a complex, but secure combination of cryptographic techniques to protect our email message.

4. Programming code

The code is developed in Microsoft Visual Studio 2010, the technology is .NET Framework and the programming language is C#.

```
class Program
{
      const int SelectCertByPosition = 0; // is first
certificate in store

      static void Main()
      {
            string strPlain = "Clear text message...";
            byte[] byteCert = null;

            // Get Certificate bytes from Personal
Certificate Store
            GetCertificateFromStore(SelectCertByPosition, ref
byteCert);
            if (byteCert == null) return;
            X509Certificate2 cert = new
X509Certificate2(byteCert);
```

1

```
            // Encrypt message in SMIME format
            CmsRecipient recipient = new CmsRecipient(cert);
            ContentInfo contentInfo = new
ContentInfo(Encoding.UTF8.GetBytes(strPlain));
            EnvelopedCms envEncrCms = new
EnvelopedCms(contentInfo);
            envEncrCms.Encrypt(recipient);
            byte[] encrResult = envEncrCms.Encode();
```

```
            // Show unencrypted message
            Console.WriteLine("Message for encrypt: " +
strPlain);

            // Show encrypted message encodet in BASE64 and
SMIME format
            Console.WriteLine("Encrypted result:");

Console.WriteLine("==========================================
===========");

Console.WriteLine(Convert.ToBase64String(encrResult));

Console.WriteLine("==========================================
===========");
```

2

```
            // Decrypt message
            EnvelopedCms encryptedMessage = new
EnvelopedCms();
            encryptedMessage.Decode(encrResult);
            // If your private is keeped on a smart card you
must enter PIN code
            encryptedMessage.Decrypt();
            byte[] result = encryptedMessage.Encode();
```

```
// Show decrypted message
            Console.WriteLine("Decrypted message: " +
Encoding.UTF8.GetString(result));
        }
        public static void GetCertificateFromStore(int 
certPositionInStore, ref byte[] x509CertRawData)
        {
            // Choice certificate store
            X509Store storeMy = new X509Store(StoreName.My, 
StoreLocation.CurrentUser); ;
            int iCount = 0;

            try
            {
                storeMy.Open(OpenFlags.ReadOnly);

                // Read certificates
                foreach (X509Certificate2 foundcert in 
storeMy.Certificates)
                {
                    if (certPositionInStore == iCount)
                    {
                        x509CertRawData = 
foundcert.GetRawCertData();
                        break;
                    }
                    iCount++;
                }
            }
            catch (Exception ex)
            {
                Console.WriteLine(ex.Message);
                x509CertRawData = null;
            }
            finally
            {
                storeMy.Close();
                storeMy = null;
            }

            return;
        }
    }
```

5. Code description

The program code described in the paper is ready for application. The purpose is to encrypt a text message in S/MIME format, applying a private key of a digital certificate, which after that will be decrypted and displayed on the screen. For the purposes of the program example Microsoft .NET Framework technology is used, where the needed cryptographic algorithms are implemented and the support for the corresponding interfaces. The programming language is C#. The code is divided in two basic blocks that are of key importance for the purpose set.

Here is their description:

1. The message encrypting is implemented in this part of the program. But it is not necessary to define the sender, who is also an owner of the digital certificate, whose keys will be used in encrypting. The certificate is defined in the object by the name “recipient”. In the variable “contentInfo” the letter content is downloaded, encoded to the corresponding code table. We have selected here UTF8, and “envEncrCms” is the class, with the help of which the content encrypting is done. This process is realized in three stages:

- the data for encrypting are downloaded with the help of an instructor at class initialization;
- the recipient is downloaded, more precisely – his digital certificate;
- at the last line of the block the message is derived encrypted, coded in the appropriate format.

In order to encrypt a message, it is not necessary to enter a PIN code, because a public key is used. This is an operation, which is accomplished by the letter’s author. The recipient is the person who decodes the message encrypted in order to read it. This process is executed in the second block, described below.

2. Decrypting of the message received. The object, defined in the code under the name “encryptedMessage” is responsible for message decoding and decrypting. At the first line we initialize it, at the next line the message is decoded with the help of a Decode method. At this stage, if the message structure is violated and not compliant with the format, the operation will be interrupted by an error message. After successful decoding of the message format, the execution passes to the following line, the decrypting operation. When the Decrypt method is invoked, the PIN code will be asked, in order to allow access to the recipient’s private key. After the PIN code is entered, the object will send the data to the crypto-processor through CSP interface. After successful decrypting of the message, it can be derived by the Encode method.

For clearer display of the program code, the selection of a digital certificate is exposed in another method, under the name “GetCertificateFromStore”. The location of digital certificates storing in Windows operation system is called a Certificate Store. The deriving of the digital certificate chosen in this case is done, reading the certificate in the respective store by its serial number. That is why the input parameters are two, the first one with the name "certPositionInStore” expects entry of the serial number, the second one – “x509CertRawData” is of reference type and serves here to store back the result – the digital certificate.

The example suggested can be easily implemented in any information system, which would considerably increase its communication security. There are some systems that require automatic notification of the user, sending them the respective information, or vice versa, in automatic process of communication – from the users towards the system. In both cases an automatic process of secured communication between the client and the server can be realized, as well as between two servers.

6. Advantages of the solution proposed

The analysis and the experiments done show that the solution proposed would reduce to reasonable limits the risk of email information frauds. The discussion of the problems and typical hackers' attacks are exposed with the purpose to give better understanding of the risks, connected with this service over-trusting.

In some cases, when the risk is evaluated, the use of a digital certificate, not protected by a smart card or any other type of specialized cryptographic module, may be accepted.

The software example applied could function with or without any additional hardware modules. It also gives a clear idea of this solution. In some cases it might prove useful in the development and protection of information systems, where the electronic messages, automatically sent, carry confidential information, which might affect the security of its users.

References

1. Bradley, T. Essential Computer Security: Everyone's Guide to Email, Internet, and Wireless Security. 2007.
2. Housley, R., T. Polk. Planning for PKI: Best Practices Guide for Deploying Public Key Infrastructure. 2001.
3. The Internet Society. RFC 3851 Secure/Multipurpose Internet Mail Extensions (S/MIME). 2004. **http://www.ietf.org/rfc/rfc3851.txt**
4. Schneier, B. Digital Security in a Networked World. 2004.
5. McNab, C. Network Security Assessment: Know Your Network. 2004.

Возможные риски для надежности email коммуникации и один подход защиты

Николай Докев, Иван Благоев

Нов болгарский университет, 1618 София
Emails: n.dokev@nbu.bg blagoev.i@gmail.com

(Резюме)

В работе описаны риски для конфиденциальности информации при использовании коммуникации путем электронной почты. Дискутируются разные способы вредительства в электронной почте и подходы их избежания и обеспечения надежности электронной корреспонденции. Предложен метод и разработан софтуерный продукт, который обьединяет преимущества наиболее часто используемых технологий. Обсуждаются области его применения.