A multidisciplinary project: how to turn a web-cam into a secure-cam

Bogdan Groza, Lavinia E. Dragomir
"Politehnica" University of Timişoara, Faculty of Automation and Computers, Department of Automation and Applied Informatics, Timişoara, Romania
bogdan.groza@aut.upt.ro, lavinia.dragomir@aut.upt.ro

***Abstract*—The paper is focused on developing a client-server application that can be used in the secure communication of video information over the IP. The main interest in developing the application is as a multidisciplinary project that can help students to develop their practical skills. The interest for such an approach in learning appears in the context of the Bologna process where multidisciplinarity seems to be the preferred learning approach at the bachelor level, while interdisciplinarity fits mostly in the higher education levels. The multidisciplinary character of the project raises from the fact that it involves knowledge from fields such as data communications, information security, image processing, computer programming, mathematics etc. Thus, implementing an application in .NET Framework and using communication protocols and cryptographic primitives to assure some security objectives is a valid response to multidisciplinary learning in engineering.**

I. Introduction

There are many prior approaches in engineering education, most of them being focused in developing platforms that can facilitate remote learning. For example, in [1] a platform for remote control is developed, while in [2] there is a similar approach for electronic circuit design. While such approaches are focused on remote learning of specific fields, we advocate that graduate engineers must have learning experiences that come from a large variety of fields. Although, finally they are going to be specialized in one field or another, acquiring a broader experience will help them in communicating with experts from other fields. More, students are acquiring mostly theoretical knowledge during lectures, but in order to consolidate the ideas it is important to gather them in practice in multidisciplinary projects.

At present, the interest of promoting multidisciplinarity and interdisciplinarity in higher education as context of training the students to approach problems that arise in practice, problems that go beyond the boundary of a single discipline, is increasing [3], [4]. Multidisciplinarity is the framework for solving a problem by putting together the contributions of researchers from different disciplines, none of the researchers going beyond the boundary of his / her own discipline. Interdisciplinarity is an approach of solving a problem in a holistic manner, by cooperation of researchers from different fields that come to intersect, generating new concepts which expand the boundaries of the involved fields and create a common conceptual framework, sometimes even novel academic fields.

In the context of the Bologna process, in which the higher education has three hierarchical levels – bachelor, master and PhD, it is important to find out to what extent multidisciplinarity and interdisciplinarity can be applied. We consider that for the bachelor level the concept of multidisciplinarity is suited and applicable, and only in singular cases the concept of interdisciplinarity fits. For the master level, both concepts can be applied. For the PhD level, also both concepts are applicable, but interdisciplinarity becomes prevailing. In this way, a graduate of all the three levels is prepared to solve practical problems that require a multidisciplinary or interdisciplinary approach.

With respect to engineering education, one way of applying a multidisciplinary approach at the bachelor level is by creating disciplines which have as purpose the development of projects from which students can acquire lessons from more than one field.

In this context, the paper focuses on the development of an application for secure remote communication of video information between multiple clients of a network (figure 1), as the theme of a multidisciplinary project which can help students in understanding security and work with some equipments and programming languages. Web-cams are used as the main image acquisition equipment. These cheap devices can be acquired from any computer shop at low costs, more, they are also used in many security applications such as surveillance tasks. Still, almost no security objective is guaranteed for the commonly used off-the-shelf cameras, and securing them can be a challenging task.

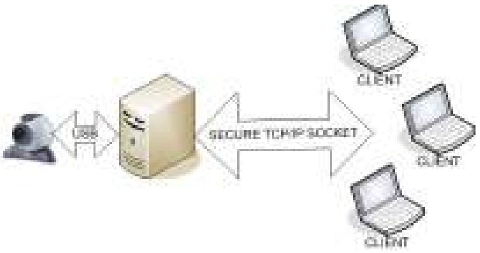

Figure 1. Application setting

The benefits in developing the proposed application are twofold: first it is useful as a project that can be developed by students in order to improve their practical skills, second it is useful as an experimental result on communication and computational performances. Our main interest here is in benefits of the first kind. The student gain in developing such an application is that it consolidates knowledge from fields such as: data communications, information security, cryptography, mathematics etc. Nevertheless, information security is a field of high interest nowadays and learning security is a topic frequently addressed. From our perspective, learning security can be done only when students confront directly with practical applications of security. Therefore, in our project we aim at letting students determine and understand what the security objectives for remote transmissions are and to develop solutions for them.

The paper is organized as follows. In section II we discuss the fields involved in the project. Section III is concerned with the implementation details. Section IV holds the conclusions of our paper.

II. Fields Related To The Project

A. *Data communications and Computer Networks*

Data communication addresses the transmission of information between two communication endpoints and is done by using digital or analogous signals. Further, interpreting these signals employs encoding/decoding techniques in time or frequency domains. One of the subfields of data communication is computer communications which addresses communication between computers that are part of a computer network. In computer networks the most popular protocol suite is the Transmission Control Protocol / Internet Protocol Suite (TCP/IP).

For facilitating practical implementations, all protocols are developed on multiple layers; the standard reference model is the Open Systems Interconnection Basic Reference Model (OSI Reference Model or OSI Model) which is based on seven distinct layers. Still, there are models that do not comply with the seven OSI layers. For example the popular TCP/IP has only four layers, while their correspondence to the OSI layers is not clearly defined. Several authors proposed different matches between the OSI layer and the TCP/IP [5], [6].

The Internet Protocol (IP) is responsible for distributing packets to the intended address on the Internet which is a packet switched network. Still this protocol cannot address the case when packets are lost or arrive in an incorrect order to the destination. For this purpose IP is augmented with the Transfer Control Protocol (TCP) which assures reliability such that packets arrive in the correct order and potential losses are detected and retransmission requested. In order to improve performance, when reliability is less important, the User Datagram Protocol (UDP) can be used. Although this protocol is not reliable, it has better performances in real-time scenarios; a good example for this purpose is a broadcast scenario.

In our platform we used the TCP/IP in order to assure a reliable communication between the endpoints. Still, students can have the opportunity to get contact during lectures with protocols such as UDP and RTP (Real-Time Transport Protocol).

B. *Information Security*

One of the most relevant issues in communication is assuring information security, which means protecting the information against communication disturbances or malicious adversaries [7]. Further, attacks due to malicious adversaries can be split in two cases: passive adversaries, who eavesdrop on the information, and active adversaries who can modify the content of the information, claim false identities etc. Nevertheless, these adversaries can be placed in a large spectrum from hackers to terrorists, while the most dangerous adversary comes from combinations between multiple types of adversaries.

Finally, security is assured with respect to security objectives. In every remote communication system there are some security objectives that must be ensured, such as confidentiality, integrity, authenticity, non-repudiation, availability, freshness, anonymity, authorization, third party protection, revocation, traceability etc. We now give an overview of the most relevant security objectives that can be discussed in the context of turning a regular webcam into a trusted webcam:

- *Confidentiality* means to assure that the video information is accessible only to authorized parties. For example, this objective is commonly present in the context of pay television etc.
- *Integrity* means to assure that the transmitted video information has not been modified during the transmission either by perturbations or by an adversary. This is a natural requirement for every transmission and is also assured by non-cryptographic meanings such as CRC codes. Also, commonly used peer-to-peer sharing applications use cryptographic hash functions for the same purpose.
- *Availability* refers to assuring that a service is accessible whenever an authorized user requests it. Achieving this objective is a fundamental issue in real-time video surveillance systems and one of the primary concerns in assuring it is the removal of potential DoS attacks.
- *Authentication* has two coordinates: first it addresses the identity of a participant, meaning the existence of a guarantee for the identity of the participant, second it addresses the information itself, meaning that the source of the information can be bind to a participant. In particular, for video information, this means that participants can validate the source of the information and if the transmitted images are altered this will be detected by them. It can be observed that by achieving authentication one also assures integrity.
- *Non-repudiation* prevents an entity to deny having sent some information. So, if a sender signs and sends some piece of video information then it cannot further deny that he actually sent it while the receiver can prove this to any neutral party.
- *Authorization* refers to the control of the access to the information and preventing the entry into a system of unauthorized parties.
- *Anonymity* means preventing the identification of a participant's identity. This means that a participant can further deny that he actually sent some information. This is useful in several scenarios such as on-line voting where users need to preserve the anonymity of their choice but one can easily extend such an objective in the context of video surveillance.

C. *Cryptography*

The most employed technique against malicious adversaries is the use of cryptography. In general, cryptography addresses the development of functions that are used in protocols in order to assure security objectives. Further, cryptographic functions can be classified in the following categories: hash functions, symmetric and asymmetric encryption functions, and digital signatures. All of the previously encountered security objectives can be assured by using cryptographic functions, and this is as well the intention for the proposed application. In this way, students get contact with modern cryptographic solutions for assuring security over the IP and have the opportunity to design their own secure socket in .NET.

One of the most unexpected field that needs to be mentioned in this context, and nevertheless important, is number theory. The relevance is that cryptographic functions make intensive use of number theory. For example the RSA cryptosystem used in our application for public key encryption involves algebraic operations over fields of integers. A complete understanding of the cryptographic operations performed with the RSA in .NET also requires some knowledge from number theory. Just to clarify the need for number theory let us enumerate the public fields of the *RSAParameters* class in .NET: *D, DP, DQ, Exponent, InverseQ, Modulus, P, Q*. The significance of these values is listed in MSDN but a good understanding of them requires certain knowledge from number theory. This is mostly because the implementation of RSA in .NET is different from the basic description of the RSA cryptosystem that can be found in most text-books in order to increase performance (this requires extra parameters such as *D, DP, DQ, InverseQ*).

D. *Computer programming*

Computer programming tends to be an important skill for engineers independent to their particular field of activity. In order to develop a complex application, basic programming skills and the ability to implement text-book algorithms are not enough. In order to deal with complex applications object-oriented programming languages must be used in which an application can be organized as a system of objects, each object being associated to a set of data and functions. Object-oriented programming has important features such as encapsulation, aggregation, derivation etc. that allow flexible implementations. In particular in our application we used Visual C# as a programming environment. The main advantages in using Visual C# are its user-friendly interfaces and its C syntax.

E. *Image processing*

Finally, another field worth to be mentioned is image processing. This is because surveillance may imply tasks from geometric transformations to color corrections of the acquired images. In the developed application, students were also asked to add basic transformations to images in order to increase their quality, for example increasing brightness or contrast. Many other image processing features can be added to the application as well.

III. Implementation Details

A. *Application architecture*

The client-server application is implemented using the Visual C# programming environment from the .NET platform. There are two distinct, stand-alone applications for the client and server parts.

Three user defined classes named *SecureSocket*, *PacketHandler* and *MACHandler* are implemented on the server side and referenced from the client side where they are needed as well. The *System.Net.Sockets.Socket* class from .NET is used for TCP/IP communication. A socket is created by instantiating the *System.Net.Sockets.Socket* class and specifying TCP as the communication protocol. The socket is associated with a local end-point – that means, the local IP address of the server and the port number of the server application - using the *System.Net.Sockets.Scoket.Bind* method of the created *System.Net.Sockets.Socket* object. The IP address of the server and the port number of the Server application are given by the user. The server socket is placed in a listening state, waiting for the client applications to make requests. An object of the *SecureSocket* class is associated to every client that requests connection to the server and the client is placed in a queue, in order to exchange data with the server. The *SecureSocket* class provides methods for receiving and sending information packets between the client and the server applications. The building of the information packets has been separated from the actions of sending and receiving them. This is done by the user defined class *PacketHandler* which has been created for building information packets and also for testing them for authenticity. The object-oriented oriented architecture of the server application is partially depicted in figure 2.

The client application uses instances of the .NET *System.Net.Sockets.Socket* class and of the *PacketHandler* class that has been defined in the server application. The Client class defines the user interface of the client application and the behaviour of the client.

Using the *System.Net.Sockets.Socket* class, a socket is created for the client in order to connect to the server. The request for connection is made using the method *System.Net.Sockets.Scoket.Connect* and giving as end-point the IP of the server (introduced by the user) and the port of the server application (also introduced by the user). The Client class provides methods for sending information packets and receiving information packets from the server. The information packets are built using the *PacketHandler* class.

B. *Assuring security through sockets – the SecureSocket class*

One of the basic functionalities of a secure transmission is authentication. It is important for students to learn that basic measures such as CRC codes are not enough when dealing with human adversaries that can easily bypass such security mechanism. In this context the main cryptographic tool for authentication are Message Authentication Codes and in our application they were used in the *MACHandler* class. Further, all the cryptographic security of the application is embedded in the *SecureSocket* class that has the following relevant methods:

- *InitializeSecureSocket* – generates the encryption keys for the socket and creates the *CryptoStreams* for reading and writing on the socket.
- *InsecureSend*, *InsecureReceive* - used to send and receive information prior to the existence of a secret key to encrypt the channel, for example to send and receive the RSA certificates.
- *SendRSAParamsAndEncryptedSecretKey* – used to send the RSA public key certificate and a fresh secret key encrypted by using the other party public key (it makes a call to *InsecureSend* as there is no secure line at this particular moment).
- *ReceiveRSAParamsAndEncryptedSecretKeyed* - used to receive the RSA public key certificate and an encrypted secret key (it makes a call to *InsecureReceive*).
- *BeginSecureReceive* – used to create a thread on which the information from the encrypted socket is read. This operation must be done on a thread in order to avoid blocking the application while reading from the socket.
- *SecureSend* – used to write information on the encrypted socket.

After exchanging the RSA parameters the server and the clients exchange a secret key with the following naïve protocol:

$$1.\mathit{Client} \rightarrow \mathit{Server} : \mathit{ClientCertificate}, \mathit{RSA}_{\mathit{Server}}(K_c)$$

$$2.\mathit{Server} \rightarrow \mathit{Client} : \mathit{ServerCertificate}, \mathit{RSA}_{\mathit{Client}}(K_S)$$

This serves only as a straight-forward example for students, but more complex protocols with provable security can be used. Finally, the client and server keys K_C, K_S are used to secure the sockets with a key derived from them in order to ensure that the key which secures the socket is pseudo-random and depends both on the key chosen by the user and the server. Further, the socket is encrypted using AES that is available on 128, 192 and 256 bits. These initialization steps are mostly done in the following function:

```
public void InitializeSecureSocket()
{
 HMACMD5 hashMAC = new HMACMD5();
 hashMAC.Key = sentKey;
 myEncryptor.Key =
      hashMAC.ComputeHash(receivedKey);
 myEncryptor.IV = IV128Bit;
 hashMAC.Key = receivedKey;
 myDecryptor.Key = hashMAC.ComputeHash(sentKey);
 myDecryptor.IV = IV128Bit;
 clientCryptoWriteStream = new CryptoStream(
      clientStream,
      myEncryptor.CreateEncryptor(),
      CryptoStreamMode.Write);
  clientCryptoReadStream = new CryptoStream(
      clientStream,
      myDecryptor.CreateDecryptor(),
      CryptoStreamMode.Read);
}
```

C. *Acquiring video information*

There are several ways in which one can acquire video information from a web-cam in Windows. Probably the most natural way is by using Windows Image Acquisition (WIA) see (http://support.microsoft.com/kb/266348) for more details. We used this library but there are also other alternatives, for example one can prefer to use the avicap32.dll library etc. Also, there is a nice example available at The Code Project based on WIA (http://www.codeproject.com/KB/cs/WebCamService.aspx). In the application we used WIA and a timer triggered an event periodically to capture images from the web-cam.

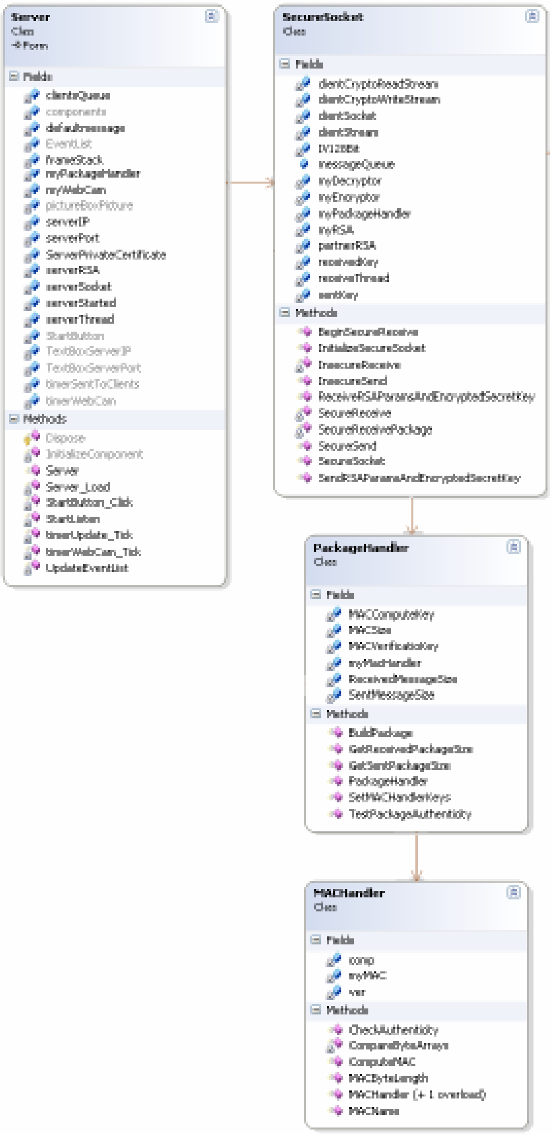

Figure 2. Application architecture

IV. Conclusions

The underlined project provides students the opportunity to acquire knowledge from several fields: data communications, computer networks, information security, cryptography, computer programming and image processing. Figure 3 describes synthetically the multidisciplinarity of the project. Besides consolidating some theoretical knowledge the students had the opportunity to develop an application that is fitted for the real world and has a practical meaning. The main focus of the developed application was on establishing security based on some cryptographic tools by the use of the implementations provided in the .NET framework. The application can be subject of further extensions and improvements as well.

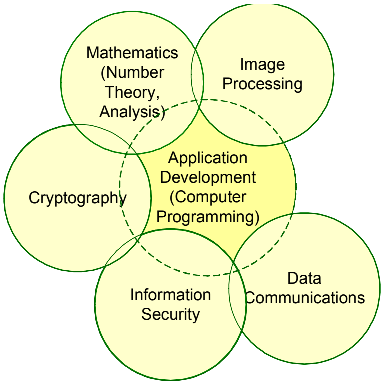
Mathematics (Number Theory, Analysis)
Image Processing
Application Development (Computer Programming)
Cryptography
Information Security
Data Communications

Figure 3. Synthetic representation of multidisciplinarity and its outcome

Acknowledgment

This work was partially supported by national university research council CNCSIS grant PNCDI PN II - 940/2009.

References

[1] J.A. Fonseca, G. Rodrigues, H. Proença, J. Rodrigues, L. Barradas, J. Ferreira, and M. Calha, "DI.SY.RE. - A Demonstrator for Distributed Industrial SYstems REmotely Controlable", *12th EAEEIE Annual Conference on Innovations in Education for Electrical and Information Engineering*, Nancy, France, 14-16 May 2001.

[2] M. Billaud, T. Zimmer, and D. Geoffroy, "The Cyberchip for analogue integrated circuit design teaching", in *Proc. EAEEIE 05, 16th EAEEIE Annual Conference on Innovations in Education for Electrical and Information Engineering (EIE)*, Lappeenranta, Finlande, 6 - 8 June 2005.

[3] E.N. Brown, "Interdisciplinary Research: A Student's Perspective", *Journal of Chemical Education*, vol. 79, No. 1 / January 2002, pp. 13-14, Published by The Division of Chemical Education of the American Chemical Society.

[4] J.M.M.C. Marques, "Inter and multidisciplinarity in engineering education", *International Journal of Mechanics and Materials in Design*, Vol. 4, No. 2 / June 2008, pp. 229-237, Springer Netherlands.

[5] W. Stallings, *Data and Computer Communications*, Pretince-Hall, Inc., ISBN-10: 8120312406, ISBN-13: 978-8120312401, 1997.

[6] A.S. Tanenbaum, *Computer Networks*, ISBN-10: 0130661023, ISBN-13: 978-0130661029, Prentice Hall, 2002.

[7] W. Stallings, *Cryptography and Network Security*, Pretince-Hall, Inc., ISBN-10: 0131873164, ISBN-13: 978-0131873162, 2005.