Dynamic Video Conferencing with Fully Secured Encryption Algorithms

Y.V. Srinivasa Murthy
Assistant Professor
Department of CSE
ANITS, VSKP

Dr. S. C. Satapathy
Professor & HOD
Department of CSE
ANITS, VSKP

Ch Rajesh
Assistant Professor
Dept. of IT
ANITS, VSKP

P Tejnadh Reddy
3/4 B.Tech
Dept. of CSE
ANITS, VSKP

ABSTRACT
This paper is mainly concerned with the internal details of a runtime platform for teleconferencing and all the study and research work done in improving the runtime platform so as to not to use it as a stand-alone application but to support reuse of its components. The platform designed has three functionalities. Converts the captured video into .x file format, Encrypt the data file and transfer it over the network and Create connections dynamically. The main objective of the paper is to facilitate a single application to handle various scenarios. Code and other resource reuse are possible for development of new applications. This concept is also economical as it reduces the number of connections at a point of time. Owing to the runtime creation of the connections, the unwanted connections are not present. The teleconferencing is networked multimedia application which requires real time audio-video streaming and collaboration among the conference participants. Tele-conference application like tele-tutoring involves many scenarios under single application, requiring an application platform to deal with the dynamic changing scenarios. The main objective of this paper is to develop a runtime platform for fast implementation of the multimedia application with conference character and collaborative feature. Dynamic change of the connection structure among the different scenarios is proposed here.

Keywords
Tele Conferencing, Encryption, Video Conferencing, Capture, Network Security

1. INTRODUCTION
Teleconference is the live exchange and mass articulation of information among several persons & machines remote form one another, but linked by tele-communications. The tele-communications system may support the tele-conference by providing one or more of the following audio, video and/or data services by one or more means such as telephone, telegraph, computer, radio and television. A teleconference is a telephone meeting among two or more participants involving technology more sophisticated than a simple two-way phone connection. At its simplest, a teleconference can be an audio conference with one or both ends of the conference sharing a speaker phone. With considerably more equipment and special arrangements, a teleconference can be a conference, called a video conference, in which the participants can see still or motion video images of each other.

An application platform is the collection of application program interfaces and protocols on which content and applications are developed. In computing a platform describes some sort of hardware architecture or software framework (including application framework), that allows software to run. Typical platforms include a computer's architecture, operating system, programming languages and related interface. A communication platform is one that supports the fast implementation of networked multimedia applications. The platform is nothing but it exhibits the notion of a site as one of its main abstractions. A site can be a collection of workstations, media input, and output devices that are, in terms of control, tightly coupled. What a platform does is it exports a programming interface with high-level abstractions for session and connection control, allowing application developers to concentrate on scenario and user interface design. The project first describes all the study done in platform architecture and programming interface, then talks about the implementation of platform and applications and their deployment in the harsh environment. The platform acts as interface between the system and network.

The platform approach allows to implement and to incrementally improve an application scenario with significantly reduced effort as compared to an approach based on stand-alone prototypes. Development platforms do exist for the creation of interactive retrieval services on residential cable networks. As the customer access link becomes symmetric, there will be a demand for multipoint and multi-user services like tele-conferences, having life-cycles maybe just as short as those of retrieval services. Development platforms for multi-user services will have to deal with dynamic connection structures and with multiple user interfaces. The proposed system features are: Eases the design and implementation of the networked multimedia applications, Supports multiple scenarios and reduces the no of connections between the conference participants. [1] [2]

The advantages of proposed System are as follows:

A. Security

This platform provides tow level security for the information transmission over the network. One is data encryption is done before transmission so as to avoid trapping of the data. Second one is that the platform provides a separate video format and its player is installed at each of the conference participant. This also authenticates the users of the platform.

B. Dynamic Nature

The platform supports dynamic connection structure which facilitates the change of the connection structure from one scenario to the other.

1.1.(Dot)X File Format

The .x file format concept is introduced so as to provide extra security to the video being transmitted over the network. The

networked multimedia applications have their prime requirement as security. A communication platform should also provide security besides providing fast and easier communication connections. The platform designed in this project provides two levels of security. The first level security is provided by encrypting the pixel information before transmitting over the network. This ensures that the video is not traced by the attackers. This encryption is done by using RSA encryption algorithm. The second level of security is provided by the .X file format defined by the platform. The platform converts the video captured from the web cam into a different file format say some .X format. The player for this multimedia file format is present only with users of the platform. The player is installed at each of the participant systems. Thus the video can only be played at the real conference participants. Even when the attacker is able to trace the video and is successful in decrypting the information, he cannot do anything with that data since the video cannot be played in any other available players. [3] [4]

This feature also authenticates the real users of the platform. That is this also acts as product key for the platform. Even though the platform software is some how managed to be obtained again nothing can be done due to the absence of the player.

1.2.RSA Algorithm

The RSA algorithm is the most popular and proven asymmetric key cryptography algorithm. In asymmetric key cryptography, also called public key cryptography, two different keys (key pair) are used. One key is used for encryption and only the other corresponding key must be used for decryption. No other key can decrypt the message- not even the original key used for the encryption. One of the keys is called public key and the other is called private key. A key pair is associated with a communicating party the encryption is done using the public key and the decryption is done by using private key. The RSA algorithm is based on the mathematical fact that it is easy to find and multiply large prime numbers, but it is extremely difficult to factor their product. The private and public keys of RSA are based on very large (made up of 100 or more digits) prime numbers. The algorithm itself is quite simple. However, the real challenge in the case of RSA is the selection and generation of the public and private keys. [5] [6]

1.3.Dynamic Connection Establishment:

Each of the system connected has two application programs

A. Server program (for speaker)

B. Client program (for client)

it is the first word of the header.)

1.1.1 Process of Dynamic Connection

- The server is executed in the system of the speaker and all other systems executes client program
- Connection is established from server to all the clients
- Clients do not have connectivity
- The clients who wants to speak send request and gets queued up
- They are given chance when the speaker is done with

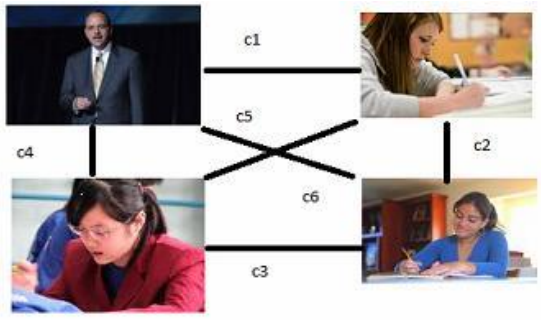

Fig 1: Process of Static Connection

The speaker executes the server program and ultimately the other users become the clients. Connections are established only from the server system to the client systems. This is because the required feature is that every conference participant needs to see the speaker and the speaker needs to look at the other participants and address them. Thus connection from the speaker to other participants is sufficient. There is no use in the existence of the connection between the non speakers since it is not used at that point of time. This project is developed mainly for the tele-teaching applications consider a professor delivering lecture which is being attended by students at three locations. Initially professor is the speaker and gives explanation over a particular topic the students are currently the clients and those who wants to speak or ask some doubt sends request to the server indicating that they want to speak. These requests are queued up in a priority queue so that all the clients get their chance according to their priority. When the professor finishes his part of the explanation then he gives chance to the student who has the 1st priority. Then the student becomes the server and the professor becomes the client. Now the connection is established from hat particular student to all other students and also the professor. Now everyone can view the student asking the question. The application scenario is now switched from classroom-professor to student-classroom. After the student finishes his question, the professor again becomes the server and the process goes on. The speaker needs to view all the conference participants so the server system is provided with screen splitting feature with each segment containing a participant video. [7]

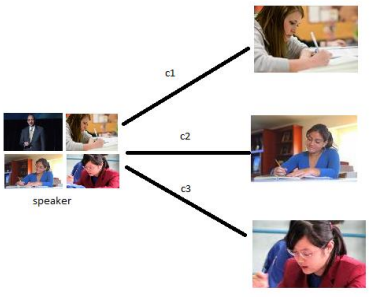

Fig 2: Process of Dynamic Connection

1.4.BETEUS Project:

The collaborative teleconferencing platform described in this article was developed in the course of the European Beteus (Broadband Exchange for Trans-European usage) project and was tested in field trials over the European ATM pilot network which interconnects the Beteus partners in France. The Beteus platform profited from the experience gained with the Betel tele-teaching application which was one of the first collaborative applications to be run over cross-national ATM links in Europe. The most important requirement put onto the platform design was a high-level API supporting connection control, session management & collaboration. The platform should also support application specific grouping of equipment so as to allow complex application endpoints like a classroom with multiple cameras and screens. The platform was to be designed on top of the TCP/UDP/IP protocol suite for portability reasons. When work on Beteus started in August 1994 it was actually not clear how the interconnecting network would look like. Another open issue was which kind of application should actually be deployed on top of the Beteus network. The principal goal of the project was to demonstrate the quality of communication that can be achieved when bandwidth is not a limiting factor. Two applications were proposed in the early Beteus deliverables: a distributed classroom application and a tele-meeting application. Rather than to start implementing an application that could become obsolete at a later stage of the project it was decided to develop an application platform with a high-level application programming interface (API) that would keep the effort to implement a specific application at a minimum. It in fact turned out that the distributed classroom application would need to be replaced by a tele-tutoring application. The two applications that were developed for the platform are a tele-meeting application for simple informal meetings, and a tele-tutoring application, which is actually a reimplementation of the Betel application on the Beteus platform. The tele-meeting application establishes all-to-all audio and video connections among its participants and allows sharing a single application. Connection management becomes only active when session vertices enter or leave the session, in which case they are automatically added or deleted from the audio, video and application sharing connections. The typical usage scenario of the tele-tutoring application is a professor tutoring a group of students, all of them in front of personal workplaces. The application has different states that correspond to the time-line of a tutoring session. First the professor gives a speech and explains the subject, in which case there is an audio and video multicast from him to the students. After that students start to work, in which case there is only a point-to-point audiovisual and application-sharing connection when the professor is answering a student question. [8]

The experience of Beteus is that the broadband network needs to be much more transparent than it is now in order to support advanced multi-point applications. The configuration effort necessary to allow for connectivity n IP level in a network as small as the one of Beteus was perceived as a major obstacle. What is needed for platforms like the one of Beteus to run successfully is signaling and network multicast support. After Beteus, the platform will be deployed on France Telecom's ATM WAN in Sophia-Antipolis that is becoming operational in April 1996. Work on it will continue in various directions. In a general move to shorten communication paths and to increase distribution within a site, we started to reemployment our platform on top of CORBA. Station agents are replaced by an object request broker that establishes direct communication between the connection manager and the audio, video and application sharing processes. A future version of the API will be defined in IDL and allow direct connection endpoint control. An application stub may then provide access to API calls and events. [9] [10] [11]

1.5.AVI Capture

This allows capture and digitization of the input analog video signals from external devices and conversion to an AVI file on the disk of the computer. No compression is applied to the video data and hence this is suitable for small files. Playback of the video is done through the Windows Media Player. Before capturing parameters like frame rate, brightness, contrast, hue, saturation etc. as well as audio sampling rate and audio bit size may be specified. [12]

AVI TO MPEG CONVERTER:

This utility allows the user to convert a captured AVI file to MPEG format. Here the MPEG compression algorithm is applied to an AVI file and a separate MPG file is created on the disk. Before compression parameters like quality, amount of compression, frame dimensions, frame rate etc. may be specified by the user. Playback of the MPEG file is done through the Windows Media Player.

MPEG CAPTURE:

Certain cards allow the user to capture video directly in the MPEG format. Here analog video data is captured, digitized and compressed at the same time before being written to the disk. This is suitable for capturing large volumes of video data. Parameters like brightness, contrast, and saturation etc. mat are specified by the user before starting capturing. [13]

DAT TO MPEG CONVERTER:

This utility converts the DAT format of a Video-CD into MPEG. Conversion to MPEG is usually done for editing purposes. DAT and MPG are similar formats so that the file size changes by very small amounts after conversion. The user has to specify the source DAT file and the location of the target MPG file.

MPEG EDITOR:

Some capture software provides the facility of editing an MPEG file. The MPG movie file is opened in a timeline structure and functions are provided for splitting the file into small parts by specifying the start and end of each portion. Multiple portions may also be joined together. Sometimes functions for adding effects like transitions or sub-titling may also be present. The audio track may also be separately edited or manipulated. [14]

MPEG-1

MPEG Stands for Moving Pictures Expert Group. MPEG-1 belongs to a family of ISO standards. Provides motion compensation and utilizes both intraframe and interframe compression. Uses 3 different types of frames: [15]

- I-frames
- P-frames
- B-fames

I-FRAMES (INTRACODED):

These are coded without any reference to other images. MPEG makes use of JPEG for I frames. They can be used as a reference for other frames.

P-FRAMES (PREDICTIVE):

These require information from the previous I and/or P-frame for encoding and decoding. By exploiting a temporal redundancy, the achievable compression ratio is higher than that of the I-frame. P-frames can be accessed only after the referenced I or P-frames has been decoded.

B-FRAMES (BIDIRECTIONAL PREDICTIVE):

This requires information from the previous and following I and/or P frame for encoding and decoding. The highest compression ratio is attainable by using these frames. B frames are never used as reference for other frames. Reference frames must be transmitted first. Thus transmission order and display order may differ. The first I frame must be transmitted first followed by the next P frame and then by the B frames. Thereafter the second I frame must be transmitted. An important data structure is the Group of Pictures (GOP). A GOP contains a fixed number of consecutive frames and guarantees that the first picture is an I-frame.

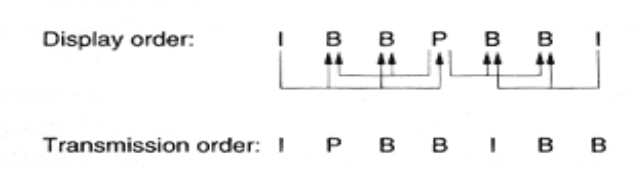

Fig 3: Frame Transmission Order

2. Analysis

Today's collaborative teleconferencing systems are usually implemented as stand-alone applications with fixed interaction and communication scenarios. They establish static audio and video connection structures among the conference participants and employ a specific tool for collaboration. The software architecture of such systems is often highly rigid; since it is designed with the requirements of a single application in mind it does not automatically support the reuse of its components within other application scenarios. This means that there is a new software design and implementation process each time a new application needs to be developed, with code reuse being at the library level or lower. It is clear that the stand alone system approach to teleconferencing application development will give way to a platform approach. Networked multimedia applications in general will be more commonly implemented on top of programming interfaces that provide different levels of control for media acquisition, transmission, and payout, or simply for whatever building block is likely to be used by a large number of applications. Some of these interfaces will be standardized, allowing applications developed or one hardware architecture to be easily ported to other ones. Authoring tools and application development platforms will further ease the design and implementation of networked multimedia applications. An application development platform is especially necessary in the case where an application is to be offered as a service in a large public or private network.

Much of the complexity there stems from the necessity to integrate the application into the network infrastructure and to make it work or coexist with other services. Development platforms do exist for the creation of interactive retrieval services on residential cable networks. As the customer access link becomes symmetric, there will be a demand for multipoint and multi-user services like tele-conferences, having life-cycles maybe just as short as those of retrieval services. Development platforms for multi-user services will have to deal with dynamic connection structures and with multiple user interfaces, just to name two sources of added complexity. The platform approach was introduced in European Beteus project whose definition of focused on the network communication aspects rather than the applications. The main requirements on the applications were that one of them be a tele-teaching application, and that they make the best use of the high bandwidth available.

Since there was no clear vision for the applications at the beginning of the project, it was decided to build an application platform rather than standalone applications for everyone of the envisaged application scenarios. The platform should constitute the highest common denominator between the envisaged application scenarios, and it should allow to implement and to incrementally improve an application scenario with significantly reduced effort as compared to an approach based on stand-alone prototypes. Thus the proposal for platform for networked multimedia application came into existence. On the similar lines

this project is also developed for tele-teaching and tele-tutoring applications with fast implementation and dynamic connection switching keeping in mind the optimality and economy involved in conducting such distributed classes

2.1 Efficient MPEG Video Encryption Algorithms

Algorithm I is a light-weight MPEG video encryption algorithm which incorporates encryption (decryption) with MPEG video compression (decompression) in one step. The primary goal of algorithm I is to save the encryption computation time by taking the advantage of combining MPEG compression and data encryption and at the same time avoid decreasing the video compression rate. We use a permutation of the Huffman codeword list as a secret key. During MPEG encoding (decoding), the encoder (decoder) uses the secret key instead of the standard Huffman codeword list. For those who do not have the secret key and use the standard Huffman codeword list to decode an encrypted MPEG video, the decoded video frames will be different from the original video frames, in order to prevent the encryption from affecting compression rate.

The advantage is that no overhead is added to MPEG codec since the encryption/decryption processes do not cost extra computation. The disadvantage is the size limitation of the key space. First, since MPEG compression rate depends on the Huffman codeword list, if we use an arbitrarily Huffman code word list to encode the MPEG video, the compression rate may decrease. To avoid affecting compression rate, we limit the permutation of Huffman codeword list (the secret key) to those code words which have the same length as the standard Huffman codeword. Second, it seems that not all of permutations of the Huffman codeword list can be used as encryption keys. This makes key generation difficult since a generated key has to be tested for validity before using.

To remedy the drawbacks of Algorithm I, a video encryption algorithm called VEA. Before describing the details of VEA, it is worthwhile to describe the general scheme of multimedia data encryption. Multimedia data encryption techniques aim to prevent unauthorized receivers from decoding the data by scrambling them. The basic principle is to apply an invertible transformation Ek1to the original data (or a portion of them).

VEA's encryption effects are achieved by the IDCT process during MPEG video decoding. Even if only some of DCT co-efficient are changed, these changes will propagate to most of IDCT coefficients. For those who have the secret key, they can decrypt the received data and get the original video. VEA's decryption function E_k^{-1} is the same as its encryption function since $E_k(E_k(S)) = S$. In VEA, encryption key and decryption key are the same. For those who do not know the secret key, their decoders will produce images very different from the original video because most image pixel values are changed. Although VEA only changes some signs of DCT co-efficient, most image pixel values will be changed during inverse DCT (IDCT) of MPEG decoding operations. The analysis shows that VEA is actually an image cipher algorithm. It shifts image pixel values according to a given secret key. VEA is much more efficient than DES algorithm because it selectively encrypts a small number of bits of the MPEG compressed video, and a selected bit is only XOR ed one me with the corresponding bit of the secret key. In contrast, DES encrypts every bit of the compressed video, and each bit is XOR ed 8 times with the given secret key plus many transposition operations. To encrypt a MPEG video, a software implementation of DES will take much more time than a software implementation of VEA does. [16]

3. DESIGN & IMPLEMENTATION

3.1 RSA Algorithm [17]

Input: encryption key (key), array pointer to plain txt(*ch), length of array(l),pi value (n).
Output: array pointer to encrypted data (*arr)

```
RSA_encryption(int key,long *ch,int l,int n )
{
//allocte memory for encrypted data (ie for arr)
w=key;i=0;

do...
b[i]=w mod 2;
w=w/2;i=i+1;
repeat till (w!=0)
k=i;
for i=0 to l
c=0;f=1;
for j=k down to 0
c=c*2;
f=(f*f)mod n;
if(b[j]=1)
c=c+1;
f=(f*ch[i]) mod n;
arr[i]=f;
return arr;
}
```

3.2 Dynamic Connection Establishment

For server:

Step 1: Start.
Step 2: If the message in BufferedReader is talk from any of the clients, do the following.
Step 3: Get the IP address of that client and store it in ip1.
Step 4: Conform dialog box is displayed.
Step 5: If the input is yes, set the status corresponding to ip1 as true and others as false.

For Client:

Step 1: Start.
Step 2: Retrieve the IP address whose corresponding status is true.
Step 3: The client socket is connected to the server socket at that particular IP address.
Step 4: When the client wants to speak, 'talk' button is clicked.

Step 5: The message is flushed on to the printwriter.
Step 6: Repeat the above steps continuously.

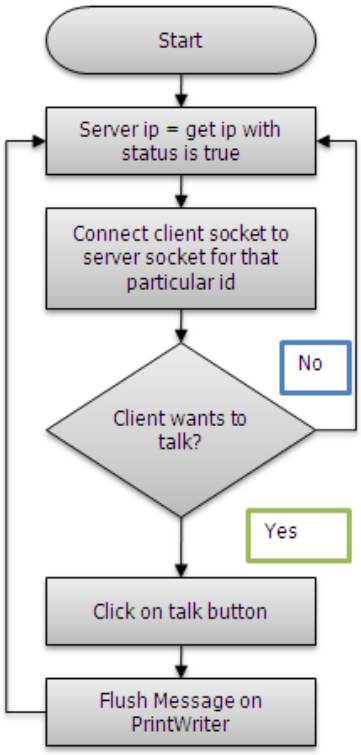

Fig 4: Flow Graph for Client - Side Operation

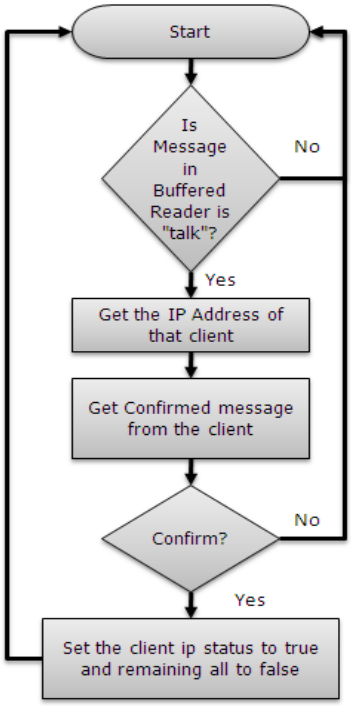

Fig 5: Flow Graph for Server – Side Operation

Steps for the video transfer:

Step 1: Capture the video at station1 by using external web cam
Step 2: Convert the video to frames
Step 3: Convert each frame to pixel files
Step 4: Encrypt the pixels to get encrypted data
Step 5: Encrypted data is transferred using print writer
Step 6: Receive the data from pipe at station2
Step 7: Read the data and decrypt the data
Step 8: convert the decrypted data to image frame
Step 9: Export the frames to video
Step10: Stream the video and display the video
Step11: Repeat the same steps at station2

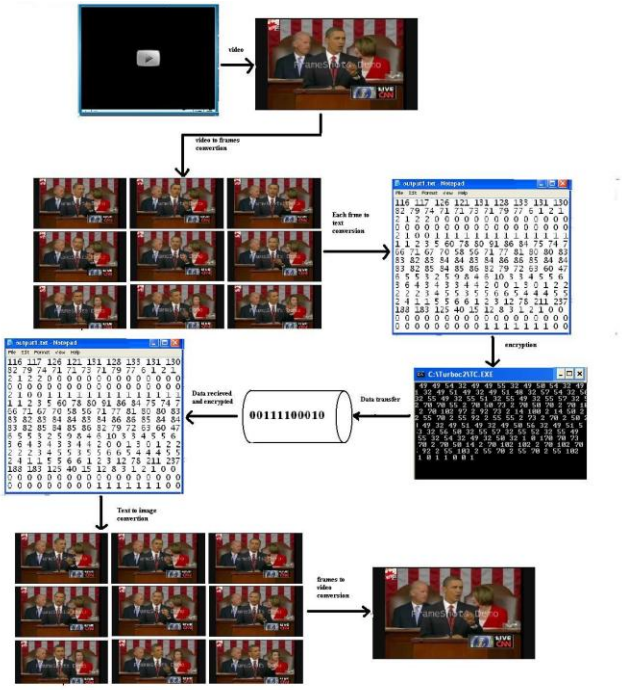

Fig 6: Video Transfer & Encryption

4. FUTURE WORK

This project "A Development to a Runtime Platform for Teleconferencing Applications" proposes to use .X file format to send the video on the network so that if any intruder attacks and sees the file then he will be able to play it. If this is implemented during teleconferencing, it provides a great deal of security while the video is being transmitted. A special video player has to be designed for this purpose only which can play these .X file format videos.

5. CONCLUSION

This project "A Development to a Runtime Platform for Teleconferencing Applications" proposes a great deal of study done on the teleconference scenarios and platforms and proposes a platform approach in which dynamic connection establishment among users i.e. when someone else other than needs to speak, the server program is made to run on his system so that he is visible to all other agents and he becomes the speaker momentarily. The server and client run on different systems employ different roles which are performed.

6. REFERENCES

[1] Chris Greenhalgh. 1997. *"Analysing movement and world transitions in virtual reality tele-conferencing"*. In Proceedings of the fifth conference on European Conference on Computer-Supported Cooperative Work (ECSCW'97), John A. Hughes, Wolfgang Prinz, Tom Rodden, and Kjeld Schmidt (Eds.). Kluwer Academic Publishers, Norwell, MA, USA, 313-328.

[2] Chris Greenhalgh and Steven Benford 1995. *"MASSIVE:A Collaborative Virtual Environment for TeleConferencing"*. ACM Trans. Comput-Hum. Interact. 2,3 (Sep – 1995), 239 – 261. DOI = 10.1145/210079.210088

[3] H. Schwarz , D. Marpe and T. Wiegand *"Overview of the scalable video coding extension of the H.264/AVC standard"*, IEEE Trans. Circuits Syst. Video Technol., vol. 17, , 2007.

[4] M. Wien , H. Schwarz and T. Oelbaum *"Performance analysis of SVC"*, IEEE Trans. Circuits Syst. Video Technol., vol. 17, p.1194 , 2007.

[5] Lawrence C. Washington and Wade Trappe. 2002. "*Introduction to Cryptography: With Coding Theory"* (1st ed.). *Prentice Hall PTR, Upper Saddle River*, NJ, USA.

[6] Johannes B, Martin Otto, and Jean-Pierre Seifert. 2003.*"A new CRT - RSA algorithm secure against bellcore attacks."* In Proceedings of the 10th ACM conference on Computer and communications security (CCS '03). ACM, New York, NY, USA, 311-320. DOI=10.1145/948109.948151

[7] Xiao L, Liu Y H and Ni M L *"Improving Unstructured peer- to-peer systems by adaptive connection establishment"* IEEE Transactions on Computers 2005, 54(9): 1091 – 1103.

[8] BETEUS, "Broadband Exchange For Trans-European Usage", Technical Annex, Project Number: M1010, 1994.

[9] BETEUS, *"Broadband Exchange For Trans-European Usage"*, Technical Annex, Project Number: M1010, 1994.

[10] BETEUS Consortium, *"BETEUS Application Platform Detailed Specification"*, Deliverable D6, November 1994.

[11] BETEUS Consortium, *"BETEUS Communication Platform Specification"*, Deliverable D5, October 1994.

[12] LIU Xiao-jun(Chinese People's Armed Police Forces Academy, Hebei Langfang 065000, China);"*Design and Realization of High Definition Video Capture Card Based on HD-SDI*";Video Engineering;2009-01

[13] Marius Preda, Blagica Jovanova, Ivica Arsov, and Franoise Prteux. 2007. *"Optimized MPEG-4 animation encoder for motion capture data"*. In Proceedings of the twelfth international conference on 3D web technology (Web3D '07). ACM, New York, NY, USA, 181 - 190. DOI=10.1145/1229390.1229425

[14] K. Brandenburg, G. Stoll, F. Dehery, and J. D. Johnston. *"The ISO-MPEG-1 audio: A generic standard for coding of high-quality digital audio"*. Journal of the Audio Engineering Society, 42(10): 780–792, October 1994.

[15] K. L. Gong, *"Parallel MPEG – 1 Video Encoding"*, MS thesis, Department of EECS, University of California at Berkeley, May 1994. Issued as Technical Report 811

[16] L. Qiao and K. Nahrstedt. *"A New Algorithm for MPEGVideo Encryption"* In Proceedings of the first International Conference on Imaging Science, Systems, and Technology (CISST'97), Pages 21 – 29, Las Vegas, Nevada, July 1997.

[17] W. S. Stallings, *"Network security Essentials"*, Second Edition, Prentice-Hall, 2000