International Journal of Computer Science and Network (IJCSN)
Volume 1, Issue 5, October 2012 www.ijcsn.org ISSN 2277-5420

# Introduction to Real Time & Secure Video Transmission using Distributed & Parallel Approach

[1]Trupti Dandamwar, [2]Manish Narnaware

[1] Department of Computer Science & Engg., G.H.Raisoni College of Engineering
Nagpur, Maharashtra, India

[2] Department of Computer Science & Engg., G.H.Raisoni College of Engineering
Nagpur, Maharashtra, India

**Abstract**

Advances in digital content transmission have increased in the past few years. However, Security and privacy issues of the transmitted data have become an important concern in multimedia technology. The paper introduces a computationally efficient and secure video encryption approach with use of distributed & parallel environment. The paper aims to make secure video encryption feasible for real-time applications without any extra dedicated hardware at receiver side.

***Keywords:*** *Video Encryption, Distributed and Parallel Approach.*

## 1. Introduction

With the technological advances in today's world, secured networked continuous media have gained utmost importance and protection from potential threats such as hackers, eavesdroppers, etc. have resulted into more research being made into making the network more secure and user friendly.

Playing video streams over a network in a real time requires that the transmitted frames are sent with a limited delay. Also, video frames need to be displayed at a certain rate; therefore, sending and receiving encrypted packets must be achieved in a certain amount of time utilizing the admissible delay. For example: Video On-Demand requires that the video stream needs to be played whenever the receiver asks for it. So, there are no buffer or playback concepts for the video stream (i.e. it runs in real time).

Thus real time & secure video transmission process is computationally intensive.

The paper is structured as follows: In Section 2, literature review is done with the brief discussion of each paper. Section 3, summarizes the discussion in brief. Section 4, introduces the Parallel and Distributed approach. Section 5 describes the proposed work using Parallel and Distributed approach and finally conclusion is drawn in Section 6.

## 2. Literature Review

The section is briefing about Literature Survey on various Video Encryption Techniques proposed by Researchers / Authors with pros and cons of each technique.

### 2.1 Evaluation of AES Encryption Technique by Wail S. Elkilani, Hatem M. Abdul-Kader [1]

The goal of this research is to focus on the following points:

- Implementing AES for MPEG-4 in a real time secure video transmitting system
- Comparing the Performance of the AES with respect to two major encryption techniques over a peer to peer channel.
- Evaluating the difference between the overhead resulting from different data types in multimedia (text, audio, and video) due to the three encryption techniques (XOR, RC4, AES)

The paper showed that the AES encryption algorithm can be used effectively to encrypt MPEG-4. The performance of AES encryption frames is sufficient to display the received Frames on time. The encryptions delay overhead using AES is less than the overhead using RC4 and XOR algorithm. Also, AES can achieve satisfactory encryption results with little overhead. Therefore, it is concluded that using AES in encrypting MPEG-4 is a feasible solution to speedy and secure real time video transmissions.

### 2.2 Light Weight Video Encryption Algorithms by Varalakshmi.L.M., Dr. Florence Sudha G. and Vijayalakshmi [2]

The paper focuses on achieving high data security at low computational time. It is achieved by encrypting the Intra frames by means of secret sharing using DCT and DWT with scrambling of motion vectors. A performance comparison based on DCT and DWT based secret sharing is done. The second proposal consists of avoiding the computationally demanding motion compensation step and tends to exploit the temporal redundancy in the video frames by transforming each group of pictures to one picture eventually with high spatial correlation and these converted Inter frames are then scrambled which effectively reduces the computational time.

By using the GF polynomial and LFSR the key space is increased. A new seed is generated for every intra frame and this makes the proposed video encryption algorithm robust to cipher text-only and known-plain text attacks. Various security measures were carried out on the new proposals and the results indicate the robustness of the proposed schemes.

### 2.3 Video Encryption Algorithm proposed by Jayshri Nehete, K. Bhagyalakshmi, M. B. Manjunath, Shashikant Chaudhari, T. R. Ramamohan [4]

The paper discusses about the algorithm selectively encrypting a fraction of the whole video. It is faster than encrypting the whole video with AES. Typically, the findings are as follows:

- MPEG-1 videos sign-bits occupy less than 10% of the entire video bit stream. Therefore it can save up to 90% of encryption time compared to the algorithm which encrypts the entire video. It encrypts at most 128 bits, no matter what type of frame is used. This considerably reduces encryption computations achieving satisfactory encryption results.
- A software implementation is fast enough to meet the real-time requirements of MPEG-1 decoding. The author believes that this can be used for secure video-on-demand applications and pay-per-view programs.

Algorithm introduced by K. Bhagyalakshmi, M. B. Manjunath using AES technique on MPEG video is as follows:

**Algorithm for Video Encryption/Decryption**

```
Begin
Open MPEG video file
Create output file
While (not end of MPEG file)
{
    Read n bytes from input file in buffer
    For each byte in buffer
    {
        If (collected sign bits == 128)
        {
            /*apply AES encryption algorithm */
            Rijndael (state, cipher_key)
            {
                Key expansion (cipher_key, expanded_key)
                add_round_key (state, expaned_key)
                /* Nr: Number of rounds,
                Nc: No. of columns of state matrix */
                For (i=1; i<Nr; i++)
                    Round (state, expaned_key + Nc*i)
                Final round (state, expanded_key+Nc*Nr)
            }
            Put resulting sign bits in original place
        }
    }
    Write n bytes from buffer to output file
}
Close input and output file
End
```

### 2.4 Parallel Multi-Key Encryption by Alexander Wong and William Bishop [3]

This paper presents an efficient parallel video encryption algorithm suitable for consumer devices. Partial video encryption techniques are used to significantly reduce the computational overhead associated with encryption while achieving an acceptable level of security. Multi-key encryption and parallel stream ciphers are used to improve both security and computational performance.

Experimental results from the encryption of various test video sequences demonstrate the effectiveness of the video encryption scheme. It is our belief that this method can be successfully implemented in low-cost consumer devices such as set-top boxes and digital movie disc players. Future work includes the design and implementation of a parallel video stream encryption processor based on the proposed algorithm.

Based on the theory presented, the encryption algorithm proposed by Alexander can be outlined as follows
(Where *n* is the number of partitions):

1. Obtain *n*+ 1 initial key and generate *n*+1 corresponding nonces.
2. Encrypt the initial keys and nonces, along with the total number of common-key groups, and store them if the video content is distributed in pre-recorded media. Otherwise, send the encrypted data over the network.
3. Combine the initial keys and nonces from Step 1 to create *n*+1 initial seeds for *n*+ 1 cryptographically secure pseudo-random number generator (CSPRNGs).
4. At the start of each common-key group, generate *n* encryption keys and one permutation key in parallel using the CSPRNGs.
5. At the start of each resynchronization group, the *n* encryption keys are used to initialize *n* stream ciphers.
6. For each frame in the resynchronization group, the data elements that need to be encrypted are selected and are divided into *n* partitions (first byte in the 1st partition, second byte in the 2nd partition, and etc.). Each partition is encrypted using the stream cipher specified by the permutation key.
   The partitions are encrypted in parallel. The data elements to be encrypted are:
   a. Sign bits of all DC coefficients
   b. Sign bits of AC coefficients at the three lowest frequencies
7. Repeat Step 6 until the entire video stream has been encrypted.

The decryption process is as follows:

1. Retrieve and decrypt *n*+ 1 random key and *n*+1 nonces along with the total number of common key groups.
2. Use the decrypted information to generate the encryption and permutation keys for all common key groups and store them into memory.
3. At the start of each common-key group, retrieve its associated encryption keys and permutation key from memory.
4. At the start of each resynchronization group, the *n* partition encryption keys are used to initialize *n* stream ciphers.
5. For each frame in the resynchronization group, the data elements that need to be decrypted are selected and are divided into *n* partitions. Each partition is decrypted using the stream cipher specified by the permutation key. The partitions are decrypted in parallel.
6. Repeat Step 5 until the entire video stream has been decrypted.

## 3. Discussion

The different techniques used for encryption of real time video are AES technique, DCT and DWT, AES technique on MPEG real time video and Parallel encryption approach for achieving security and performance.
The paper proposes the new concept of video transmission using Parallel and Distributed approach. The concept behind distributed approach is to make real time video transmissions faster.

## 4. Concept of Video Transmission using Parallel and Distributed approach

The purpose of introduction of parallel and distributed approach is to transmit the data at faster pace and without compromising security. The technique and protocol used for transferring video and audio data which make video transmission much secure and fast is described in this section. Protocol TAPI is used for transferring audio video data securely.

### TAPI

As telephony and call control become more common in the desktop computer, a general telephony interface is needed to enable applications to access all the telephony options available on any computer. The media or data on a call must also be available to applications in a standard manner.
TAPI 3.0 provides simple and generic methods for making connections between two or more computers and accessing any media streams involved in that connection. It abstracts call-control functionality to allow different, and seemingly incompatible, communication protocols to expose a common interface to applications.
IP telephony is poised for explosive growth, as organizations begin a historic shift from expensive and inflexible circuit-switched public telephone networks to intelligent, flexible, and inexpensive IP networks. Microsoft, in anticipation of this trend, has created a robust computer telephony infrastructure, TAPI. Now in its third major version, TAPI is suitable for quick and easy development of IP telephony applications.

### Inside TAPI 3.0

TAPI 3.0 integrates multimedia stream control with legacy telephony. Additionally, it is an evolution of the TAPI 2.1 API to the COM model, allowing TAPI applications to be written in any language, such as C/C++ or Microsoft® Visual Basic®.

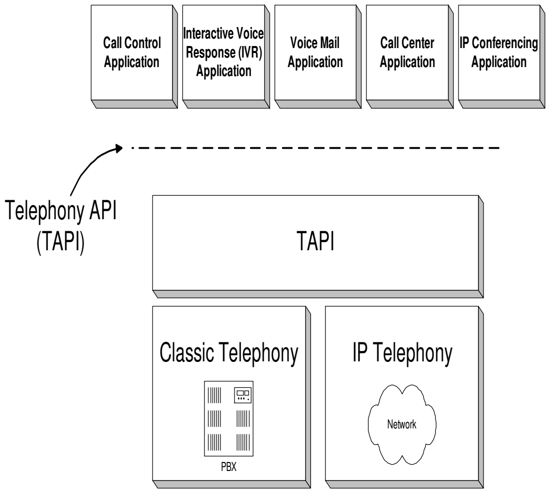

Fig. 1 Convergence of IP and PSTN telephony

Besides supporting classic telephony providers, TAPI 3.0 supports standard H.323 conferencing and IP multicast conferencing. TAPI 3.0 uses the Windows® 2000 Active Directory service to simplify deployment within an organization, and it supports quality-of-service (QoS) features to improve conference quality and network manageability.

There are four major components to TAPI 3.0:

- TAPI 3.0 COM API
- TAPI Server
- Telephony Service Providers
- Media Stream Providers

In contrast to TAPI 2.1, the TAPI 3.0 API is implemented as a suite of COM objects. Moving TAPI to the COM model allows component upgrades of TAPI features. It also allows developers to write TAPI-enabled applications in any language.
The TAPI Server process (TAPISRV.EXE) abstracts the TSPI (TAPI Service Provider Interface) from TAPI 3.0 and TAPI 2.1, allowing TAPI 2.1 Telephony Service Providers to be used with TAPI 3.0, maintaining the internal state of TAPI.
Telephony Service Providers (TSPs) are responsible for resolving the protocol-independent call model of TAPI into protocol-specific call-control mechanisms. TAPI 3.0 provides backward compatibility with TAPI 2.1 TSPs. Two IP telephony service providers (and their associated MSPs) ship by default with TAPI 3.0: the H.323 TSP and the IP Multicast Conferencing TSP, which are discussed below:

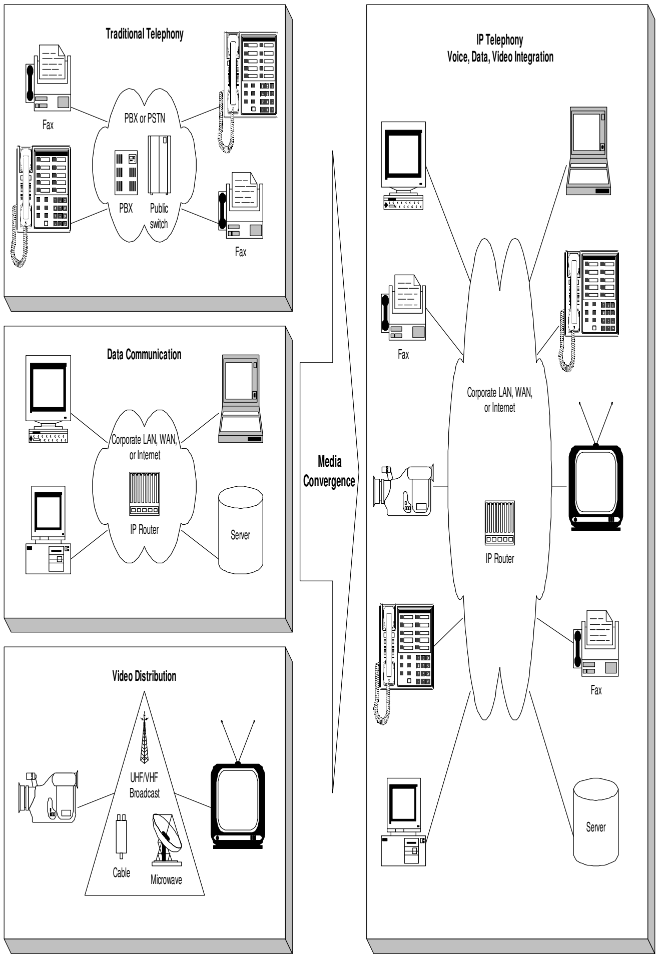

Fig. 2 TAPI architecture

TAPI 3.0 provides a uniform way to access the media streams in a call, supporting the DirectShowTM API as the primary media-stream handler. TAPI Media Stream Providers (MSPs) implement DirectShow interfaces for a particular TSP and are required for any telephony service that makes use of DirectShow streaming. Generic streams are handled by the application.

Another fast transmission technique which is latest and differ from all other technique for making parallel and distributed approach must faster and secure are as follows:

## 1. Open MPI

Open MPI represents the merger between three well-known MPI implementations:

- FT-MPI from the University of Tennessee

- LA-MPI from Los Alamos National Laboratory
- LAM/MPI from Indiana University

With contributions from the PACX-MPI team at the University of Stuttgart. These four institutions comprise the founding members of the Open MPI development team.

These MPI implementations were selected because the Open MPI developers thought that they excelled in one or more areas. The stated driving motivation behind Open MPI is to bring the best ideas and technologies from the individual projects and create one world-class open source MPI implementation that excels in all areas. The Open MPI project names several top-level goals:

- Create a free, open source software, peer-reviewed, production-quality complete MPI-2 implementation.
- Provide extremely high, competitive performance (low latency or high bandwidth).
- Directly involve the high-performance computing community with external development and feedback (vendors, 3rd party researchers, users, etc.).
- Provide a stable platform for 3rd party research and commercial development.
- Help prevent the "forking problem" common to other MPI projects.
- Support a wide variety of high-performance computing platforms and environments.

## 2. Open MP

OpenMP is an implementation of multithreading, a method of parallelizing whereby a master *thread* (a series of instructions executed consecutively) *forks* a specified number of slave *threads* and a task is divided among them. The threads then run concurrently, with the runtime environment allocating threads to different processors.

The section of code that is meant to run in parallel is marked accordingly, with a preprocessor directive that will cause the threads to form before the section is executed. Each thread has an id attached to it which can be obtained using a function (called omp_get_thread_num ()). The thread id is an integer, and the master thread has an id of 0. After the execution of the parallelized code, the threads join back into the master thread, which continues onward to the end of the program.

By default, each thread executes the parallelized section of code independently. Work-sharing constructs can be used to divide a task among the threads so that each thread executes its allocated part of the code. Both task parallelism and data parallelism can be achieved using Open MP in this way.

The runtime environment allocates threads to processors depending on usage, machine load and other factors. The number of threads can be assigned by the runtime environment based on environment variables or in code using functions

## 5. Proposed Work

The intent of the proposed work is discuss the use parallel and distributed approach for data transmission.
The proposed work utilizes the following approach for real time secure video transmissions:

- Video to be divided into frames
- Divided frames have to be encrypted
- Send the encrypted frame on the network and Collect the encrypted frames.

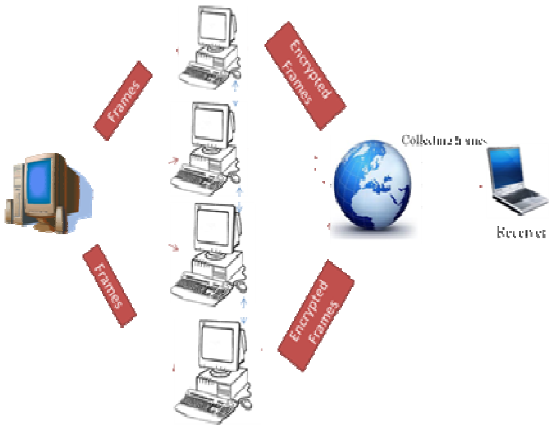

Fig. 3 Architecture of distributed and parallel approach

The architecture of proposed work is as follows:

For encryption of a video, video data will be compressed by video compression algorithms and then video will be divided into say m sets of n number of frames. It may be possible that more than one frame are same in a set of n frames. These m sets will be distributed among p machines in a cluster. Each machine in the cluster will encrypt the distinct frames only. A machine will be using parallel approach for same. The encrypted frames will be then sent to the network, according to their respective rank. Collection and reordering of the frames will be done on receiver's side machine. Receiver will have to enter the key in order to decrypt the video.

Hence, in the network, the receiver who has key will only decrypt the video making the transmission process more secure.

Another advantage of the approach is that the speed of transmission is increased due to the data is processed over series of computers rather than relying on one computer.

The flow design of the proposed work shown as follow:

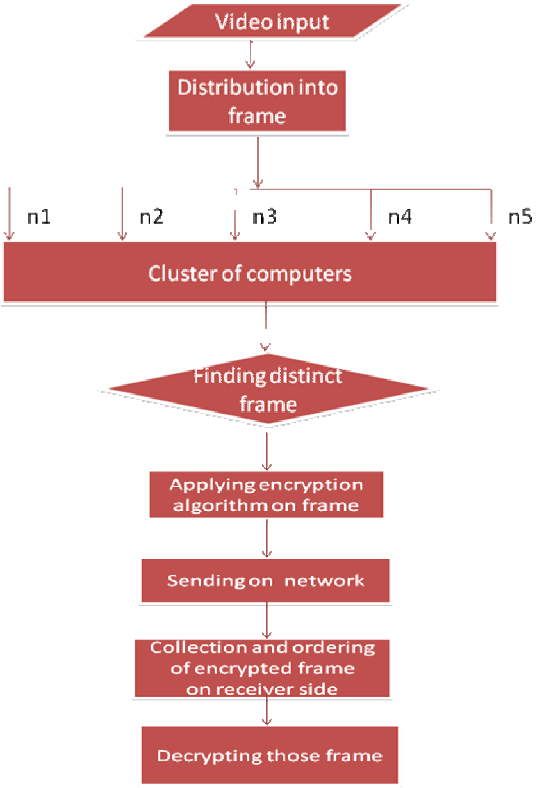

Fig 4: Flow diagram of proposed work

## 6. Conclusion

The paper has discussed various Encryption methods available for real time video transmission. Currently there is no solution and implementation of secure and fast video transmission by using distributed and parallel approach. The paper has introduced the new approach by using latest tool called Open MPI and the Protocol TAPI.

## References

[1] Wail S. Elkilani, Hatem M. Abdul-Kader Faculty of Computers and Information, Minufya University, *IEEE* 2009, pp 130-134

[2] Varalakshmi.L.M. Dr. Florence Sudha. G. Vijayalakshmi. V, Associate Professor/ Dept. of ECE. Proceedings of 2011 International Conference on Signal Processing, Communication, Computing and Networking Technologies (ICSCCN 2011)

[3] Alexander Wong and William Bishop Department of Electrical and Computer Engineering, University of Waterloo Waterloo, Ontario, Canada, 2005

[4] Jayshri Nehete, K. Bhagyalakshmi, M. B. Manjunath, Shashikant Chaudhari, T. R. Ramamohan Central Research Laboratory, 2005

[5] Jolly shah and Dr. Vikas Saxena IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 2, March 2011 ISSN (Online): 1694-0814 www.IJCSI.org

[6] ZHENG Ji-ming, GAO Wen-zheng. Color image encryption algorithmbased on chaotic map. Computer Engineering and Design, 2011, pp.2934-2937

[7] Jay M. Joshi, Kiran R. Parmar and Upena D. Dalal, "Design and Implementation of KASUMI Algorithm in ISMACryp Encryption for Video Content Protection in DVB-H Application", *IEEE International Conference on Control, Robotics and Cybernetics (ICCRC 2011), vol 1, pp 18-21, March 2011.*

[8] M. Abomhara, Omar Zakaria and Othman O. Khalifa, "An Overview of Video Encryption Techniques", *IACSIT International Journal of Computer Theory and Engineering, Vol. 2, No. 1, pp 103-110, February,* 2010.

[9] Fuwen Liu, Hartmut Koenig. "A survey of video encryption algorithms", Journal of Computers and Security, pp 3-15, 2010.

[10] Z. Shahid, M. Chaumont and W. Puech, "Fast Protection of H.264/AVC by Selective Encryption", *WSPC – Proceedings: Singaporean-French IPAL Symposium, SinFra 2009, Fusionopolis, and September 2009.*

[11] Ouni. T, Ayedi. W and Abid.M, "New low complexity OCT based video compression method", Proceedings of international Conference on

Telecommunications, Marrakech, Morocco, pp.202-207, July, 2009.

[12] Shiguo Lian, Dimitris Kanellopoulos, and Giancarlo Ruffo, “Recent Advances in Multimedia Information System Security,” International Journal of Computing and Informatics, Vol. 33, No.1, 2009, pp. 3-24.

**Trupti Dandamwar** Trupti is undergoing her Maters Degree in Computer Science and Engineering in G H Raisoni College of Engineering, Nagpur. She has completed her undergraduate degree in year 2011 from Rajiv Gandhi college of Engineering and research technology with First Class. Her research interests are Artificial intelligence and Distributed & parallel processing.

**Manish Narnaware** He has completed his Maters Degree in year 2010 from dept. of Computer Science and Engineering, VNIT Nagpur, with first class. Undergraduate degree in year 2002 from VNIT Nagpur. He has around 4 years of professional experience. His research interests are distributed & parallel processing, Computational Mathematics. Best paper published by him is “practical approaches of image encryption/scrambling using 3D Arnolds Cat map” on CNC 2012, Springer Link Digital Library.