IJFEAT

INTERNATIONAL JOURNAL FOR ENGINEERING APPLICATIONS AND TECHNOLOGY

Real Time & Secure Video Transmission Using OpenMPI

Ms. Gunjan Lonarkar[*1],Prof.Yogadhar Pandey[2]

[*1]*Computer Science and Engineering, Sagar Institute of Research & Technology,bhopal Madhya Pradesh, India*
lonarkar.gunjan@gmail.com

[2]*Computer Science and Engineering, Sagar Institute of Research & Technology,Bhopal Madhya Pradesh, India*
p_yogadhar@yahoo.co.in

Abstract

*The wide use of digital images and videos in various applications brings serious attention to the security and privacy issues today. Many different encryption algorithms have been proposed in recent years as possible solutions to the protection of digital images and videos, among which MPEG videos attract most attention due to its prominent prevalence in consumer electronic markets. 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 transmission approach with use of distributed & parallel environment. The paper aims to make secure video transmission feasible for real-time applications***.**

Index Terms: *Video Encryption, Distributed and Parallel Approach, Secure Video Transmission.*

1. Introduction

Advances in multimedia technologies have popularized applications like video conferencing, pay-per-view, Video-On Demand, video broadcast, etc. In such applications, confidentiality of the video data during transmission is extremely important. This necessitates secure video encryption algorithms. While some of the encryption methods that are used to secure and protect a transmission media from potential threats such as hackers, eavesdroppers, etc. have resulted into more research being made into making the network more secure and efficient.

In a real time environment playing video streams over a network 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 with following contents: In section II, literature review on all the techniques of video encryption has been discussed. Section III summarizes the discussion in brief. Section IV presents introduction about Distributed & parallel approach and section V gives conclusion about advanced methodology that can be used for better outcome.

2. LITERATURE REVIEW

2.1 AES Encryption Technique:

AES is a key-iterated block cipher. The algorithm is same at encryption and decryption side except at the time of decryption, inverse operations are performed. The input to the cipher is one dimentional array of plaintext which is converted as state matrix. For each round, transformation round key is derived using cipher key and never specified directly. Each round transformation is composed of four different transformations such as ByteSub, ShiftRow, MixColumn, and AddRoundKey. The repeated application 10 rounds of transformation is performed on the state. 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

- 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)

2.2 Light Weight Video Encryption Algorithms

The research work emphasize on following goals:

- To achieve 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.
- Focus on avoiding the computationally demanding motion compensation step and tends to exploit the temporal redundancy in the video frames.

2.3 Encryption Algorithm based on Hyper-Chaos

The paper discusses about the algorithm based on hyper-chaos which apply the chaotic map to image and video encryption because of its unpredictability, sensitive reliability on initial value and so on a range of traits.

At beginnings, four chaotic sequences are generated from the hyper chaotic lattices presented in formula (1), which have high initial sensitivity and good randomness; then the sequences are used to encrypt the DC and AC coefficients of reference frames and scramble the motion vector of predictive frames. The simulation results and performance analysis indicate that this algorithm has a large enough key space to resist attacks, high speed encryption and good real time behavior.

2.4 A Real-Time Video Encryption Exploiting the Distribution of the DCT coefficients

The paper work focuses on achieving better transmission throughput with reduced video size. In this paper proposed algorithm is fast, making it suitable for real-time applications, yet possessing practically acceptable levels of security without much overhead on the MPEG encoding and decoding process. The basic idea of this algorithm is to perform encryption followed by permutation of the DC and AC coefficients based on the statistical properties of the DCT coefficients. This results in a minimal increase of the video size. The algorithm takes an average encryption time of 7.2 milliseconds per frame, making it ideal for real- time encryption.

2.5 Parallel Multi-Key Encryption Technique

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.

3. Discussion

As per above discussion the different techniques used for encryption of real time video are AES technique, DCT and DWT, & parallel multi-key technique. By overall observation it can be seen that AES technique seems to be more effective on MPEG real time video and Parallel encryption approach. Still there is a trade-offs when applying different encryption algorithms to MPEG video stream and its choice rely on the achieving security and performance for an applications. 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.

The paper introduces a computationally efficient and secure video encryption approach with use of distributed & parallel environment.

4. Proposed Methodologies

Distributed and parallel approach for video transmission leads to faster video sharing without compromising security. The technique and protocol used for transferring video and audio data which make video transmission more secure and faster 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

ISSN: 2321-8134

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®.

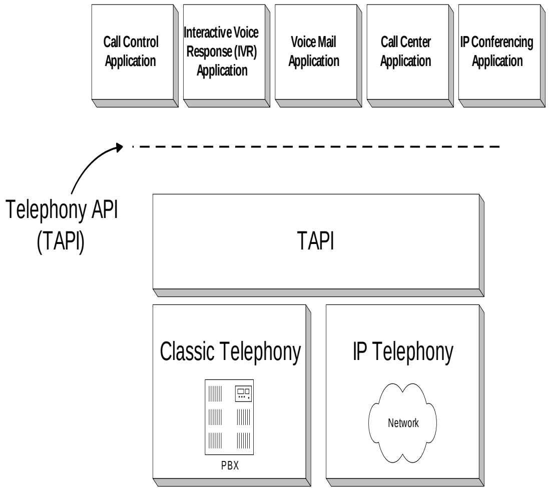


Fig. 1 Convergence of IP and PSTN telephony

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

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:

4.1 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.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.

4.2 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.

System Architecture

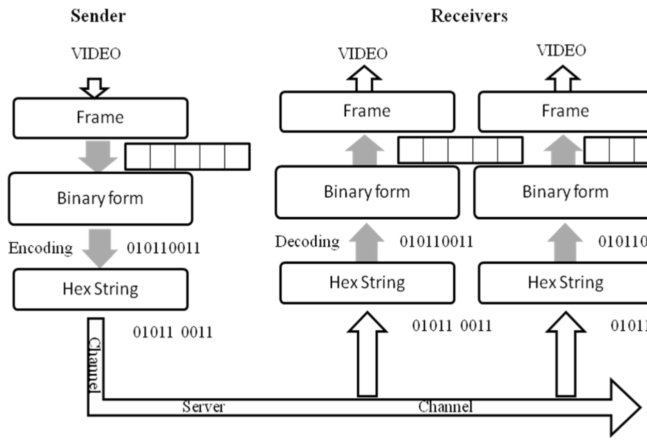
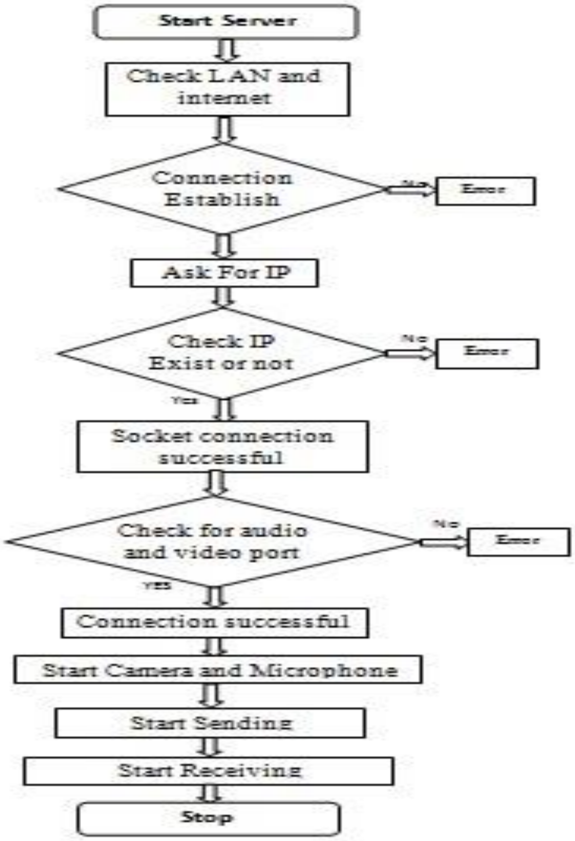


Fig. 2 System Architecture

Computer is a digital electronic device which uses binary encoder. Whenever sender sends video to the receiver, the video is mainly divided into frames (120 frames / sec) then, frames are converted into binary format i.e. 0100011000100. This is a 13 bit binary number. It takes bit of time to transmit 13 bit binary number. Open MPI first establishes connection between two computers and then start transferring the data, 'MPI_Comm_connect' function establishes connection with a server specified by port_name. 'MPI_Comm_accept' allows communication with the receiver. Many programs are written with the master-slave model, where one process plays a supervisory role, and the other processes serve as compute nodes. In this framework, 'MPI_Comm_size' and 'MPI_Comm_rank' are useful for determining the roles of the various processes of a communicator. When connection establishes through server's channel, Open MPI converts that binary bit to HEX string i.e. 0100 0110 0010. It is only 3 bit string. Now it is easy to transfer and require less time as compared to 13 bit.Open MP provide multiple threads. If the number of threads has not been explicitly set by the user, the default is implementation-defined by 'omp_get_num_threads'. In above figure two threads are created. So instead of sending video one by one Open MP sends it in parallel which gives fast transmission.At receiver's side Open MPI performs an exactly opposite job. It converts HEX string to binary bit and then decoder decodes binary bit to frames the continuous motion of frames is nothing but a video.



. Fig. 3. Flow diagram of Real time video Transmission

Thus,once the connection with the server is established the program establishes connection with server IP address. Subsequently, the process of enabling the different ports with server starts. When the computer is connected with the server, the sender can send the videos, pictures, frames with voice communication to receiver. Start camera button will used to enable capturing photos. This provides option for communication by video. . In order to establish the connection with receiver, the receiver has to input the service IP address along with the password generated by sender. Once these details are entered, the connection is established with sender and receiver can now receive the video or frames sent by sender. The receiver needs to enable microphone to establish audio communication with sender.

5. Conclusion

The project introduced and implements the new technique i.e. Open MPI and Open MP for parallel and distributed approach, by considering video conferencing as an application. The project has discussed and used the Protocol TAPI and various methods available for real time video transmission. The new technique gives secure and fast video transmission without any extra hardware efforts at receiver end.Open MPI is the new implementation of the MPI standard. It provides functionality that has not previously been available in any single, production quality MPI implementation, including support for all of MPI-2, multiple concurrent user threads, and multiple options for handling process and network failures.

References

1. ZHENG Ji-ming, GAO Wen-zheng. Colour image encryption algorithm based on chaotic map. Computer Engineering and Design, 2011, pp.2934-2937.
2. 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.
3. 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.
4. 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.
5. Uhl, A., Pommer, A.: Image and video encryption: from digital rights management to secured personal communication. In: Advances in Information Security, vol. 15. Springer, New York(2009)
6. 29. Wu, H., Bao, F., Deng, R.H.: An efficient known plaintext attack on fea-m. In: Proceedings of the Fifth International Conference on Information and Communications Security (ICICS 2008).
7. 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.
8. D. S. Abd Elminaam, H. M. Abdual Kader, and M. M. Hadhoud, “Evaluating The Performance of Symmetric Encryption Algorithms”, International Journal of Network Security, Vol.10, No.3,PP.216–222, May 2010.
9. Abdul Monem S. Rahma , and Basima Z.Yacob“The Dynamic Dual Key Encryption Algorithm Based on joint Galois Fields”, International Journal of Computer Science and Network Security, VOL.11 No.8, August 2011.
10. Manzoor Elahi, Xinjie Lv, Wasif Nisar, Imran Ali Khan, Ying Qiao Hongan Wang,” DB-Outlier Detection Algorithm using Divide and Conquer approach over Dynamic DataStream” International Conference on Computer Science and Software Engineering 2008
11. R. C. Gonzalez and R. E. Woods, “Digital Image Processing”, Second Edition, Printice Hall Inc, 2002.
12. Iain E. G. Richardson. "H.264 and MPEG-4 Video Compression" The Robert Gordon University, Aberdeen, John Wiley & Sons Ltd, UK, 2003.
13. Li & Drew," Fundamentals of Multimedia ", Chapter 5, Prentice Hall 2003.
14. Salah Aly. A Light-Weight Encrypting For Real Time Video Transmission. Available from http://www.cdm.depaul.edu/research/Documents/TechnicalReports/2004/TR04-002.pdf. (Accessed on March 2, 2009).
15. S. Lian, Multimedia Content Encryption: Techniques and Applications. CRC, 2008.
16. C.-P. Wu, C.-C. J. Kuo, “Design of integrated multimedia compression and encryption systems,” IEEE Trans. Multimedia, vol. 7, no. 5, pp. 828-839, 2005.
17. Adam J. Slagell. Known-Plaintext Attack Against a Permutation Based Video Encryption Algorithm. Available from http://eprint.iacr.org/2004/011.pdf. (Accessed on March 2, 2009).
18. V. Talwar, S. Kumar Nath, K. Nahrstedt, RSVP-SQoS : A Secure RSVP Protocol, in Proc. of IEEE International Conference on Multimedia and Expo 2001 (ICME2001), Tokyo, Japan, August, 2001.
19. J. G. Apostolopoulos, W. Tan, S. J. Wee, Video Streaming: Concepts, Algorithms, and Systems,HP Laboratories Palo Alto, Hewlett-Packard, sept, 2002
20. C. Wang, H. B. Yu and M. Zheng, “A DCT-based MPEG-2 Transparent Scrambling Algorithm,“ IEEE Transactions on Consumer Electronics, Vol. 49, No. 4, Nov. 2003.