IP MULTICAST VIDEO BROADCASTING SYSTEM WITH USER AUTHENTICATION

Hiroki Onishi
Kyoto University
Graduate School of Engineering
Kyoto Daigaku Katsura, 615-8510 Japan

Tetsutaro Uehara
Kyoto University
Graduate School of Engineering
Kyoto Daigaku Katsura, 615-8510 Japan

Takashi Satoh
The University of Kitakyushu
Faculty of Environmental Engineering
1-1 Hibikino, Wakamatsu-ku, 808-0135 Japan

Katsunori Yamaoka
Tokyo Institute of Technology
Graduate School of Science
O-okayama 2-12-1-S3-68, 152-8552 Japan

ABSTRACT

This report describes a pay broadcasting system for the Internet. This system would enable tens of thousands of people to access an identical video stream simultaneously. In this proposed system, contents are broadcast to all terminals using IP multicast. Contents are encrypted so that legitimate users can decode them with a private key and session keys. As a key management scheme, the Tracing Traitor scheme is adopted because it offers advantages in scalability. The system can also embed digital watermarks, which act as a psychological deterrent to illegal copying and distribution of copyrighted contents. Finally, implementation of an application system is described and efficient broadcasting of contents with this system is demonstrated.

1. INTRODUCTION

Recently, multimedia streaming on the Internet has become commonplace concomitant with the spread of high-speed access services such as ADSL and fiber-to-the-home (FTTH). Internet-based content distribution systems are now providing new business opportunities.

Most of the present Internet broadcast systems distribute contents using unicast. To allow numerous clients to access identical content simultaneously, the broadcast server must be a distributed database. Consequently, content providers suffer from high costs including introduction and maintenance.

Given this background, this paper discusses the necessary components to construct an Internet-based pay broadcasting system and a realistic design for such a system. In developing this architecture, we specifically addressed scalability to a large number of users and a mechanism to protect copyrighted contents. We then implemented a preliminary system based on the proposed architecture, and verify efficiency of the contents transmittion.

2. TARGET ARCHITECTURE

In this section, we consider the architecture of the Internet-based pay broadcasting system proposed in this paper.

2.1. Number of users and contents

The system is designed to support situations in which a number of users on the order of ten thousand can all obtain near-simultaneous access to the same video content.

Contents are distributed in the system as video streams. In addition, we assume that the system can support live broadcasting.

2.2. Broadcasting infrastructure

Either a distributed database or multicast communication can be considered to achieve higher scalability. For the distributed-database approach, some distribution systems apply content delivery networks (CDNs). A CDN, however, imposes a heavy cost on the content distributor, including the introductory cost, because servers must be set up at various places in the network.

For the multicast-communication approach, models include IP multicast, peer-to-peer (P2P) methods, and satellite Internet. IP multicast has not yet been established as a broadcast infrastructure because of the difficulty of routing. To solve this problem, source-specific multicast (SSM)[1] specialized for one-to-many communication has been proposed. SSM offers the potential to be easily implemented and to become popular. The P2P distribution model is excellent in terms of scalability and relatively low introductory cost, but standardization still requires much time. Currently, almost no applications use satellite Internet because the introductory cost is high. For that reason, it has not spread much.

Our system is based on the assumption of a one-to-many broadcast infrastructure using one of the methods above. The remainder of the paper uses IP multicast as an example.

2.3. Security

Implementation of a pay broadcasting system requires restriction of unauthorized reception and content duplication. Contents are distributed by IP multicast. Therefore, they must be transmitted while applying cryptographic techniques.

To restrict reception, keys are managed using a sophisticated scheme in which only authorized users can use the keys. To decrease costs and support various devices connected with the Internet, the keys are not distributed as hardware as are IC cards. Instead, they use software tamper-resistant mechanisms.

0-7803-9332-5/05/$20.00 ©2005 IEEE

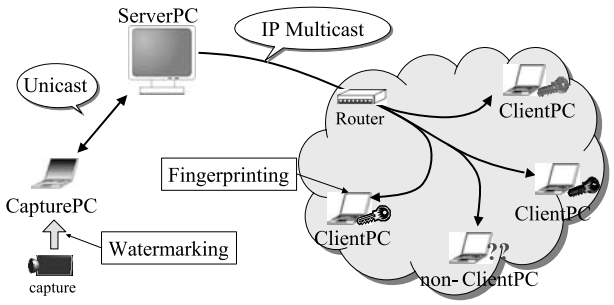

Fig. 1. System model

In addition to providing outstanding quality, digital contents can be copied without degradation. As a result, there is a large risk of illegal copying. Therefore, it is necessary to take measures to protect copyrights and prevent illegal copying. Our system solves this problem by applying digital watermarking.

2.4. System model

As a result of the above discussion, the system model that we assume is illustrated in Fig. 1. Contents are offered from a *CapturePC*; then they are broadcast to all terminals using IP Multicast while being processed on real time. The CapturePC only offers content. A *ServerPC* manages user information and employs a broadcasting system.

3. SECURITY FOR PROPOSED SYSTEM

This section describes the methods of user authentication that are utilized in the proposed broadcasting system.

3.1. User authentication

3.1.1. Selection of encryption method

Once a server transmits content over a distribution channel, that content is received by all users: both authorized and unauthorized users. Therefore, in a pay broadcasting system, the server must encrypt the content. We call the encryption key used in such a system a "session key." The server publishes a key for each client. Thereby, only authorized users are able to decrypt contents. A very simple scheme is to give each user the same key. With this approach, however, a data provider cannot identify sources of leakage of keys to illegitimate receivers. Instead, the data supplier in our system gives each user a different key – a "personal key" – thereby deterring leakage. Each user computes a session key using a personal key and then decrypts contents using the session key.

Appropriate key management schemes include Group Key Management Protocol (GKMP)[2, 3], Broadcast Encryption[4], and Tracing Traitor[5]. Elsewhere [6], we compared these schemes in terms of the amount of data transmission, the tolerance for collusion, and other factors. Table 1 shows the result of the comparison.GKMP, which is discussed elsewhere in greater detail[2, 3], must initialize a session key individually for each user who participates in a session. Therefore, the server may be overloaded in cases where many users participate in the same session. We infer that GKMP is not applicable in the proposed system because of its lack of scalability. Comparison of Broadcast Encryption with Tracing Traitor shows that the latter scheme is superior in terms of the personal key size and tolerance for collusion. Hence, as a key management scheme, this paper assumes the use of Tracing Traitor.

Table 1. Comparison of key management schemes

scheme	GKMP	Broadcast Encryption	Tracing Traitor
scalability	×	◯	◯
personal key size	◯	△	◯
cost to distribute session key	◯	×	×
tolerance for collution	◯	×	△

3.1.2. Tracing Traitor

In this scheme, the data supplier generates a base set of random keys and assigns subsets of those keys to users as personal keys. Different personal keys may have a nonempty intersection. A session key is distributed by multicasting encrypted values under all keys of the base set. Every authorized user is thereby able to compute the session key from these encrypted values using a personal key.

As an example, we describe the simple case of n users.

1. The server generates $2 \log n$ random keys. These keys are organized in a matrix with $\log n$ rows and two columns.
2. Each user has a $\log n$ bit user ID. According to the user ID, one key is selected per row of the matrix. The set of $\log n$ keys is the personal key for a user.
3. When a session begins, a session key is encrypted under the keys of the matrix; it is broadcast to all users.
4. Each user receives the encrypted session key. He or she selects elements of the matrix according to the user ID, then decrypts the session key.
5. Each user can reproduce the content using the session key.

This scheme has several variations. In a secret one-level scheme, it is possible to detect at least one traitor if the base set of random keys is a matrix with $\frac{4}{3}klog(n/p)$ rows and $4k$ columns where n denotes the number of users, k denotes an upper bound on the number of traitors, and p denotes the probability of false identification.

3.2. Deterrence of unauthorized duplication

Utilizing the Tracing Traitor scheme, we can prevent traitors from distributing their decryption keys to other users. We cannot, however, prevent illegal duplication and redistribution of contents. Digital watermarking is an effective means of solving this problem.

Our system can embed watermarks, which utilize the signature of the copyright holder at encryption and a fingerprint associated with each user at decryption, in digital contents. As a result, decrypted contents are different in each user. To prevent leakage of content that has not yet had watermarks embedded, it has been proposed to combine processes of fingerprinting and decryption at the receiver[7]. Our system assumes applications of this same technique. This paper does not describe the watermarking technology in detail.

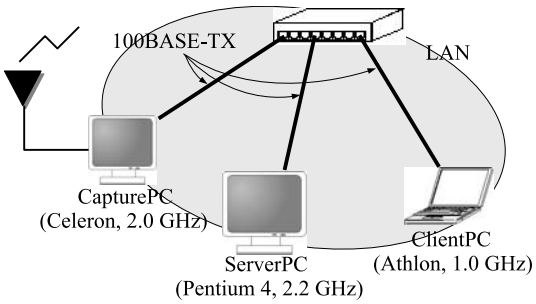

Fig. 2. Model of implemented system

4. IMPLEMENTATION

Based on the considerations discussed above, we implemented an Internet-based pay broadcasting system as an experiment.

4.1. System structure

The model of our system is illustrated in Fig. 2. We implemented the system using only software, without the watermarking technology. Computers in this experimental system were in the same VLAN.

A television broadcast stream was used as the content. The CapturePC captured it and transmitted it to the server. The ServerPC received the data, encrypted it, and distributed it using IP multicast at 500 kbps. At the same time, the ServerPC also distributed a session key encrypted according to the Tracing Traitor scheme. The *ClientPC* received a personal key from the ServerPC beforehand. Then, it received the encrypted content and the session key, and reproduced the content.

4.2. Process of the CapturePC

The CapturePC captures the television broadcast stream and transmits it to the ServerPC. At that time, communication between the CapturePC and the ServerPC requires information exchanges, such as the identity of the transmitter. Therefore, we define VSUP (Video Streaming Uploading Protocol) as an application protocol. The VSUP commands are shown in Table 2. The ServerPC judges whether the requirement is possible, and replies to the commands.

In VSUP, the CapturePC establishes a TCP connection to the ServerPC, and exchanges information on the connection. VSUP has functions, such as user authentication, specifying a data format, and transfer of the contents. VSUP has a state, called a test mode, such that the CapturePC cannot transmit content in real time. In the test mode, the CapturePC can check network speed. In addition, the CapturePC does not close the connection as soon as all of the contents are transmitted, but confirms whether a broadcast ends.

4.3. Process of the ServerPC

Before a session begins, the ServerPC generates a matrix of random keys and distributes user IDs and personal keys utilizing that matrix. We implemented the Tracing Traitor scheme using $n = 65,536, k = 10, p = 2^{-14} \simeq 6.1 \times 10^{-5}$. Therefore, the matrix of random keys has 400 rows and 40 columns. In addition, each matrix element has a value of one bit, for simplicity.

Table 2. VSUP commands

command (option) $\cdots$	meaning
HELO (IP address) (protocol) (version)	start session
USER (user name)	supply user name
PASS (password)	supply password
TEST	enter test mode
UPLOAD (data form) (bitrate) (framerate)	supply data form
TESTSEND (byte)	send test data
TESTEND	end test mode
USERAUTH (group name)	specify group
CHANNEL (send form) (IP address) (port)	specify channel
SEND (frame number) (byte)	send data
END	end sending data
PING	ping
CONFIRM	confirm end
QUIT	end session

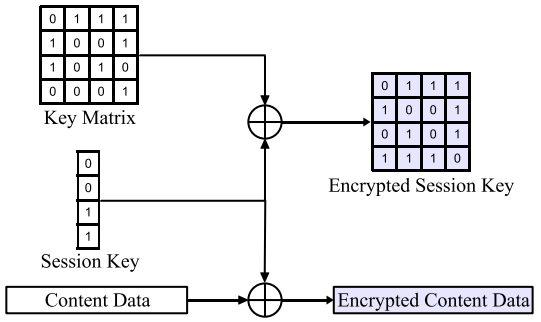

Fig. 3. ServerPC process

While contents are broadcast, the server randomly generates a session key. The session-key length is 50 bytes. An encrypted session key is the bitwise XOR of the session key and the matrix of random keys. A packet of the encrypted session key (a session packet) is distributed once every 100 packets.

The ServerPC encrypts a data packet, which is provided from the CapturePC, under the session key, and multicasts the packet. Its size is 2,048 bytes. The encryption method is as follows. A hash function is chosen. The session key is used as an input. Encrypted data is the bitwise XOR of the hash value and the content data. The ServerPC process is illustrated in Fig. 3.

4.4. Process of the ClientPC

After the beginning of a session, the ClientPC waits for the session packet. When it is received, the ClientPC decrypts the session key according to the Tracing Traitor scheme.

The decrypted content is the bitwise XOR of the content data and the hash value, which is the same one that the server used. The ClientPC process is illustrated in Fig. 4.

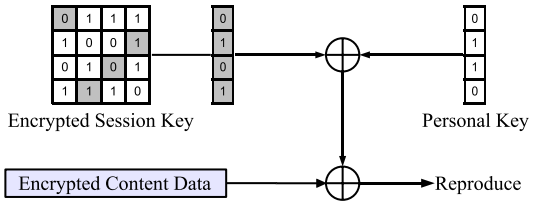

Fig. 4. ClientPC process

4.5. System evaluation

Through this implementation, we verified that the system can function in real time if a bitrate is less than 1.44 Mbps. If the bitrate is higher, the CapturePC cannot operate accurately.

Next, we describe the result at 500 kbps. CPU loads of each PC were 55 % at the CapturePC, 2 % at the ServerPC, and 15 % at the ClientPC. There is a room for adding the embedding process for watermarking. A packet loss on the network seldom occurs because the system was used in the same VLAN. A series of four packets was dropped in 100,000 packets. We intend to recover this using error-correcting code.

The time until the session packet was received and the number of discarded data packets was checked after the client program began. Results are shown in Fig. 5. The session packet should be received within 3.25 s, according to calculated values, because the bitrate of contents is 500 kbps, the size of the data packet is 2,048 bytes, and the session packet is distributed once each 100 packets. However, the influence of buffering showed that it takes slightly longer. Because about 0.3 are required for processing that actually reproduced content after reception of the session packet as a result of measurement, each user was able to reproduce contents within about 4 s.

The matrix of random keys had 400 rows and 40 columns, so the system was inferred to support 65,536 users and tolerate collusion of at most 10 users. To confirm that inference, the following was done.

1. choose 10 users at random, and generate an invalid personal key from theirs
2. reverse a bit of the matrix of the encrypted session key, and distribute the matrix as the session packet
3. if the decoder cannot reproduce the content or if the video is obviously confused, mark the users who choose the element
4. the user with the largest number of marks is exposed as the traitor

This processing was performed 50,000 times. Results successfully indicated the traitor in all cases.

Table 3 shows the sizes of the matrix of random keys, the session key, the user ID, and the personal key. Because the size of the data packet was 2,048 bytes and the session packet was distributed once each 100 packets, the data transfer amount related to the distribution of the session key was suppressed by about 1% of the overall traffic. In addition, the data transfer amount concerned with the distribution of the user ID and the personal key was 450 bytes per user. The entire data transfer amount was about 30 Mbytes. This presents no difficulties in the system because the web server, which communicates at a speed of 10 Mbps, can reasonably be expected to transmit 30 Mbytes per minute.

Table 3. Sizes of the keys

Key Matrix	Session Key	User ID	Personal Key
2000 (bytes)	50 (bytes)	400 (bytes)	50 (bytes)

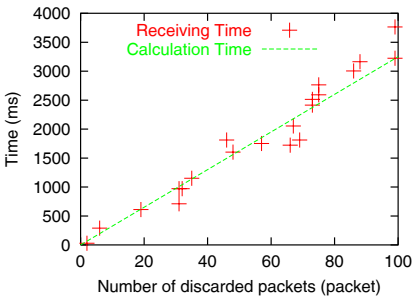

Fig. 5. Receiving Time

5. CONCLUSION

Based on a discussion of the amount of network traffic, the security level of encryption, and the processing time, we have demonstrated that an Internet-based pay broadcasting system can be implemented efficiently using IP multicast for distribution, the Tracing Traitor scheme for user authentication, and digital watermarking to protect copyrighted contents.

As a subject for further research, we will seek to verify the practicality of the system by experimenting in a wider-area environment. We also want to concretely specify the watermarking method, evaluate the system performance in terms of such factors as the quality of streaming, speed, and robustness, and implement a practical system.

6. REFERENCES

[1] H. Holbrook and B. Cain, "Source-Specific Multicast for IP," *Internet Draft*, 2003.

[2] H. Harney and C. Muckenhirn, "Group Key Management Protocol (GKMP) Specification," *RFC2093*, 1997.

[3] H. Harney and C. Muckenhirn, "Group Key Management Protocol (GKMP) Architecture," *RFC2094*, 1997.

[4] A. Fiat and M. Naor, "Broadcast Encryption," *Proc. Advances in Cryptology-Crypt'93*, pp. 480–491, 1994.

[5] M. Naor and B. Pinkas, "Threshold Traitor Tracing," *Lecture Notes in Computer Science*, vol. 1462, pp. 502–517, 1998.

[6] T. Uehara, R. Kawakita, Y. Tsuji, T. Satoh, K. Yamaoka, Y. Izumi, S. Saito, Y. Kunieda, and K. Yuki, "An Internet Broadcasting System with User Authentication on IP Multicasting," *J. of IPSJ*, vol. 44, no. 3, pp. 610–624, 2003.

[7] D. Kundur and K. Karthik, "Video Fingerprinting and Encryption Principles for Digital Rights Management," *Proceedings of the IEEE*, vol. 92, pp. 918–932, 2004.