不正複製抑止機構を備えたユーザ認証つき IPマルチキャスト映像放送システムの設計

大西 宏樹 † 上原 哲太郎 † 佐藤 敬 ‡ 山岡 克式 §

概要

同時視聴者数が数万人規模の有料のインターネット放送システムを実現するための機構について考察した．通信にはIPマルチキャストを用い，コンテンツは暗号化して送信する．正規ユーザはこの暗号化に用いた鍵を使用することによりコンテンツの視聴が可能となる．また，正規のユーザが受信したデータを不正に流用することを防ぐために，本システムでは発信時に著作権者の署名としての電子透かしをコンテンツに埋め込んでから送出し，さらに受信時にユーザ情報の電子透かしを埋め込む．これにより不正コピーに対しての心理的抑止や，著作権の主張といったことが可能となった．以上の考察をもとに実際にシステムを一部実装し，動作を確認することにより，そのようなシステムが実現可能であることを示した．

Design of IP multicast video broadcasting system with user authentication equipped with mechanism for deterrence of unauthorized duplication

Hiroki Onishi† Tetsutaro Uehara† Takashi Satoh‡ Katsunori Yamaoka§

Abstract

A pay broadcasting system for the Internet is described. That is to enable tens of thousands of people to access the same video stream simultaneously. In the proposed system, contents are encrypted so that legitimate users can decode them with a private key and session keys and broadcast them to all terminals by using IP multicast. The system can also embed ”watermarks,” which utilize the signature of the copyright holder at encryption and a fingerprint associated with each user at decryption. This acts as a psychological deterrent to the illegal copying and distribution of copyrighted contents. Finally, the implementation of an application system is described, and efficient broadcasting of content with this system is demonstrated.

1. はじめに

近年インターネットは広帯域化が進んでおり，ADSLやFTTH(Fiber To The Home)などのサービスの普及により，企業だけでなく一般家庭においても広帯域接続が実現されつつある．これにともない，インターネット上で長時間にわたって映像や音楽を流し続けるマルチメディアストリーミング通信が行われるようになってきた．今後は家庭においてもこれらマルチメディアストリーミング通信のアプリケーションを利用したり，テレビなどの音楽や映像などのコンテンツを提供するビジネスがインターネットを介して実現されるようになると思われる．

一方現在，人々の嗜好の多様化によりテレビなどは多チャンネル化が進んでおり，安価に実現できるインターネット放送ではさらに多チャンネル化が進むと考えられる．これにより，1チャンネルあたりの視聴者数が減少し，広告収入により無料の放送ビジネスを行うことは困難になると予想される．よって，インターネット放送においては，課金をおこなったユーザに対してコンテンツを視聴する権利を与える，有料放送が中心となると思われる．

このような背景により，インターネットを介した有料放送をどのように実現するか，といった研究が現在求められている．

現在，インターネットを使った映像や音楽の放送は，数十〜数百kbps程度に圧縮された映像をユニキャストを用いて伝送するシステムがいくつか実用化されている．ユニキャストを用いた通信の場合，クライアントか

†京都大学大学院工学研究科
‡北九州市立大学国際環境工学部
§東京工業大学学術国際情報センター

らの要求に応じてコンテンツを配信するという，いわゆる「オン・デマンド」サービスが可能である．しかし，1対1通信であるためサーバから送出されるデータストリーム数が同時接続されたクライアント数に比例して増加し，クライアント数に対するスケーラビリティを確保できないといった問題がある．

このような背景により，本論文ではインターネット有料放送の構築に必要となる要素技術と，実用化可能なシステム構成について考察した．本論文では前提条件として，ユーザ数に対するスケーラビリティが確保できるということ，放送ビジネスにおいて大きな問題となるコンテンツを保護する仕組みがあるということ，という2点に重点を置いて議論した．そして，実際にシステムを開発した．

2. 対象とするアーキテクチャ

ここでは，本論文で提案するインターネット有料放送システムが対象とするアーキテクチャについて述べる．

2.1 コンテンツの種類と配信方式

本システムで配信するコンテンツは動画像や音声といったストリーム通信向けのデータである．また，ライブ放送にも対応できるシステムとする．

コンテンツの配信方式としては，事前にコンテンツをサーバに蓄積し，スケジュールに従い配信するといった方式と，コンテンツをリアルタイムに処理しながら同時に配信する方式が考えられる．前者の方式では「オン・デマンド」による通信が可能であるが，サーバのストレージ容量により番組の長さに制約を受けることや，ライブ放送には対応できないといった問題がある．本システムではスケーラビリティを確保するため，またライブ放送に対応できるシステムを想定しているため，後者の配信方式を採用する．

2.2 ユーザ数と課金方式

現在のインターネット放送は主にユニキャストによる通信のため，例えば数万人のような多数のクライアントが同時に同一のコンテンツを見るためにはサーバを冗長化，分散化する必要があり，コンテンツ配信者に大きな負担となる．本論文では既存のインターネットインフラストラクチャを用い，コンテンツ配信者に負担をかけることなく，数万人から数十万人のユーザが同時に同一のコンテンツを視聴可能なシステムを目指す．

課金方式としては，課金によりある一定期間の視聴のための権利となる鍵を受け取る，いわゆる月単位契約といった方式と，ユーザの視聴時間に比例して料金を支払う方式の2通りが考えられる．本システムでは，ユーザ数に対するスケーラビリティを確保するために前者の方式であるいわゆる月単位契約方式を採用する．この方式では一度契約すると課金単位の途中ではその契約を無効にすることはできない．ただし，課金単位の途中からのシステムへの参加は可能であるとする．これにより後に述べるユーザ認証機構を簡潔にする．

2.3 放送インフラストラクチャ

現在行われているマルチメディアストリーミング通信の問題点であるスケーラビリティの確保を実現するための方法として，サーバを冗長化，分散化する方法や，データグラムを一対多伝送する方法が考えられる．前者はCDN(Contents Delivery Network)と呼ばれる方式があり現在いくつかの放送型の配信において用いられている．しかし，この方法ではサーバをネットワーク上の様々な場所に設置しなければならず，導入にかかるコストの面なども含めてコンテンツの供給者に対して大きな負荷となる．

後者はIPマルチキャストや，SHOUTcast[1]やPeerCast[2]のようなP2Pを利用したもの，衛星インターネットを用いる方法などがある．IPマルチキャストは経路制御の困難さなどから，現在世界中に放送できるインフラストラクチャとしては整っていない．経路制御の問題の解決策として，一対多通信に特化したSSM[3](Source Specific Multicast)といったものが提案されている．SSMに関しては比較的簡単に実用化が可能となると考えられ，今後普及が進むと期待される．またIPv6へ移行するにあたってIPマルチキャストが標準機能として実装されることから，経路制御の問題についても改善される可能性がある．IPマルチキャストはこのような流れから今後普及が進み，一般利用が可能になると考えられる．

P2Pフレームワークによるデータ配送はスケーラビリティに優れ，導入コストも抑えられるが，動画配送などに関する標準化などはまだ行われておらず今後の普及には時間がかかると思われる．

衛星インターネットを用いた方式は，アプリケーションがほとんどないことや受信を開始する際にかかる初期費用が高いことなどから，あまり普及していない．

本システムではこれらいずれかの方式により，データグラムが一対多伝送できる放送インフラストラクチャが整っていることを前提にする．以下の議論では，このうちIPマルチキャストを例に議論する．

2.4 セキュリティ

有料コンテンツ放送ビジネスを実現するためにはまず，受信を制限することと，受信後のコンテンツの複製を制限することが必須である．

受信の制限として，正規ユーザ以外はコンテンツを視聴不可能とし，また，正規ユーザが故意に非正規ユーザに対して視聴を手助けするような行為の抑止も必要である．配信にIPマルチキャストを用いるため，コンテンツはそれ自体を暗号化して伝送する必要がある．そこで，その暗号化に用いる鍵をなんらかの方法で正規ユーザのみが使用できるように配布することで受信を制限する．ここで，本システムではインターネットにつながる広範囲な機器をサポートしたいことやコストを下げたいといった理由から，鍵をICカードなどのハードウェアとして配布するのではなく，ソフトウェアとして実現する．

また，デジタルコンテンツはその優れた品質の他に，いくらコピーしても品質が劣化しないという特徴がある．この特徴から，デジタルコンテンツは容易に不正コピーされてしまうリスクを負っている．そこで，著作権の保護，主張，不正コピーの防止といったことが必要となる．このような問題を解決する手段として電子透かし[4]が有効である．

3. 放送システムの実現

ここでは，前節で述べたシステムを実現するためのユーザ認証方式と不正複製抑止方式について述べる

3.1 ユーザ認証方式

3.1.1 暗号化手法の選択

IP マルチキャストのトラフィックは認証なしに受信できるため，有料放送として実現するためにはコンテンツ自体を暗号化して配信する必要がある．本システムではリアルタイムストリーム通信を対象とするので，処理速度の速い共通鍵方式を用いてコンテンツを暗号化する．共通鍵方式とは，サーバとクライアントで共通の鍵を使用する方式である．以後，この鍵のことをセッション鍵と呼ぶ．サーバは各クライアントに対して鍵を発行することにより，正規ユーザのみがデータを復号できるようにする．もし，全てのユーザに同じ鍵を発行したならば，ある正規ユーザがその鍵を正規ユーザでない者に受け渡したとしてもどのユーザが鍵を漏洩したのか追跡することができない．そこで，各ユーザにはそれぞれ異なる鍵(以後認証用鍵と呼ぶ) を配布し，鍵の漏洩を抑止する．各ユーザは与えられた認証用鍵を用いてセッション鍵を取り出し，そのセッション鍵を用いてコンテンツを復号，再生する．これを図 1 に示す．

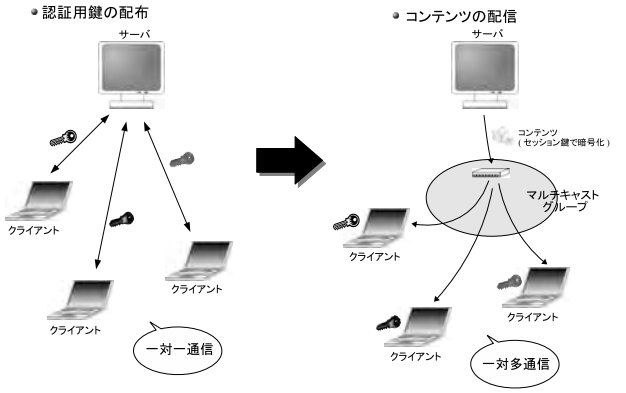

図 1: 認証用鍵の配布とコンテンツの配信

現在セッション鍵の管理方式として，GKMP(Group Key Management Protocol)[5, 6]，放送型暗号 [7]，Tracing Traitor 方式 [8] といったものがある．これらをデータ転送量や結託に対する耐性などについて比較したものが参考文献 [9] にあり，まとめると表 1 のようになる．詳しい説明は参考文献 [5, 6] にあるが，GKMP はセッションに参加してきた各ユーザに対し個別にセッション鍵を初期化する必要がある．そのため，一度に多数のユーザがセッションを開始した場合にサーバに対して大きな負荷がかかり，スケーラビリティに欠けることより，本システムが想定する規模に対応できないと考える．また，放送型暗号と Tracing Traitor 方式を比較すると，認証用鍵のサイズ，結託への耐性の面で Tracing Traitor 方式のほうが優れていると考え，本論文ではセッション鍵の管理方式として Tracing Traitor 方式を採用する．

表 1: セッション鍵配布方式の比較

方式	GKMP	放送型暗号	Tracing Traitor
スケーラビリティ	小	大	大
認証用鍵サイズ	小	大	小
セッション鍵配送コスト	小	大	大
ユーザ結託の耐性	強固	弱	中

3.1.2 Tracing Traitor

この方式はサーバで鍵の集合を用意し，その中から各ユーザにあらかじめそれぞれ異なる部分集合を配布する．セッション鍵の配布時にはサーバは同一の暗号化されたデータをマルチキャストにより配信し，各ユーザはあらかじめ配布された鍵を用いて復号する．

原理

Tracing Traitor の例として，n 人までのユーザをサポートする場合を説明する．

1. n 人までの各ユーザに配布する認証用鍵を作るため，まずサーバで $2\log_2 n$ 個の鍵の行列を生成する．これを図 2 のような行列であらわす．
2. 各ユーザに 2 進数の ID 番号を付与し，それに基づいてこの $2\log_2 n$ 個の鍵の行列から各行について ID 番号に対応した鍵を取り出すことにより，$\log_2 n$ 個をそれぞれのユーザの認証鍵として配布する．たとえば ID 番号が {0,1,1,...,0} のユーザは $\{a_1^0, a_2^1, a_3^1, ..., a_{\log_2 n}^0\}$ の鍵を受け取る．こうしてそれぞれのユーザは異なる認証用鍵を受け取る．
3. セッションが開始されると，サーバは鍵の行列を用いてセッション鍵の行列をユーザに送信する．このときのセッション鍵を $S = \{s_1, s_2, ..., s_{\log_2 n}\}$ とすると，$a_j^0, s_j \Rightarrow e_j^0$，$a_j^1, s_j \Rightarrow e_j^1 (1 \leq j \leq \log_2 n)$ としてセッション鍵の行列を図 2 のように生成する．
4. 受信した各ユーザは，暗号化されたセッション鍵の行列からユーザ ID，認証用鍵を用いて実際のセッション鍵を取り出す．たとえば ID 番号が {0,1,1,...,0} のユーザはセッション鍵の行列から $\{e_1^0, e_2^1, a_3^1, ..., e_{\log_2 n}^0\}$ の鍵を取り出し，あらかじめ配布されている認証用鍵を用いて復号し，セッション鍵を得る．
5. 各ユーザはこのセッション鍵を用いてコンテンツを復号することができる．

以上が簡単な説明である．しかし，このままではユーザ同士の結託があった場合にサーバのもつ鍵の行列が判明してしまう場合がある．上の例ではユーザ ID が排他的論理和となっている二人のユーザの結託によりサーバの持つ鍵の行列が判明してしまう．そうなると，任意のユーザ ID の認証用鍵を不正に生成することができる．また，任意の 2 名のユーザが結託すると，これらの認証用鍵を合成して新たに認証用鍵を不正に発行できる．

実システムへの応用

実際には，鍵の行列の行と列を広げ，また，サーバの鍵の行列から各ユーザが取り出す位置をユーザ ID に適当なハッシュ関数を用いて決定するようにして使用する．こうすることにより，サーバの鍵の行列が判明することは困難になり，また，ユーザ同士の結託により生成された不正な認証用鍵から結託に関わったユーザを追跡できるような仕組みとなっている．

この方式では，サーバの鍵の行列を $4k^2 \log_2 n$ 行 $2k^2$ 列とすることにより，ユーザ数 n に対して k 人までが結託しても追跡することができる [8]．ここで，k は任意に決めることのできる定数である．k を大きくするほど結託は困難になるが，セッション鍵のサイズが大きくなってしまうので，用途に合わせて適切な値を選択する必要がある．

また，この方式では複数のユーザが認証用鍵を部分的に共有しており，あるユーザがセッションの途中で脱退したくしても，そのユーザの鍵のみを無効にすることが困難である．一人のユーザの鍵を無効にするためでさえ全ユーザに鍵を配り直すことが必要となる．頻繁に認証用鍵の再配布が行われることはトラフィックの増加とつながり望ましくないので，本システムでは先に述べたように課金方式をいわゆる月単位契約方式とし，課金単位ごとにのみ認証用鍵の再配布を行う．

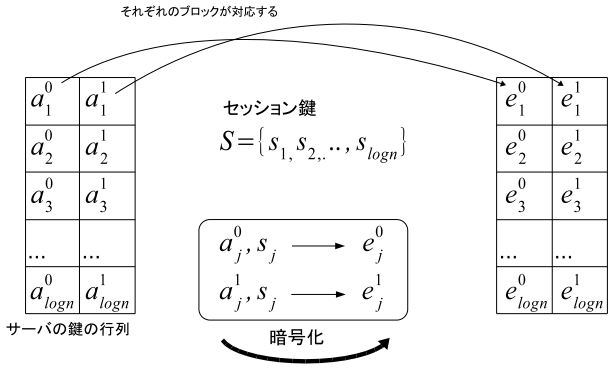

図 2: セッション鍵の生成

3.2 不正複製抑止方式

Tracing Traitor 方式を用いたユーザ認証を用いることで，鍵の不正な受け渡しは抑止できる．しかし，正規ユーザが受信したコンテンツを不正コピーし，流用することについてはこのままでは防ぐことができない．このような問題を解決する手段として電子透かしが有効である．本論文では実際の手法には言及せず，本システムに適用する際の課題について述べるにとどまる．

3.2.1 本システムにおける電子透かしの課題

本システムではコンテンツの作者が自分の署名として埋め込む電子透かし (watermarking) と，再生中のコンテンツに受信者のユーザ情報を埋め込む電子透かし (fingerprinting) の 2 種類の電子透かしを用いる．ユーザ情報を埋め込む前のコンテンツの漏洩を防ぐ方法として，暗号化の解除時にユーザ情報が残るような電子透かし手法が，参考文献 [10] に提案されており，本システムでも同様の方法を用いることを前提としている．

まず，一般に言えることであるが電子透かしは頑健であり取り外すことができないことが要求される．また，本システムではライブ放送への対応や，再生しながらの埋め込みを考えているのでリアルタイムに電子透かしを埋め込むことができる必要がある．つまり，電子透かしを埋め込む際に生じる計算量を抑える必要がある．

本システムではコンテンツに受信者のユーザ情報を電子透かしとして埋め込むために，電子透かし埋め込み後のコンテンツが復号したユーザごとに異なる部分が発生する．そのため，ユーザ間のコンテンツデータの比較によって電子透かし情報がどの部分に埋め込まれているかということが判明する可能性がある．これを図 3 に示す．

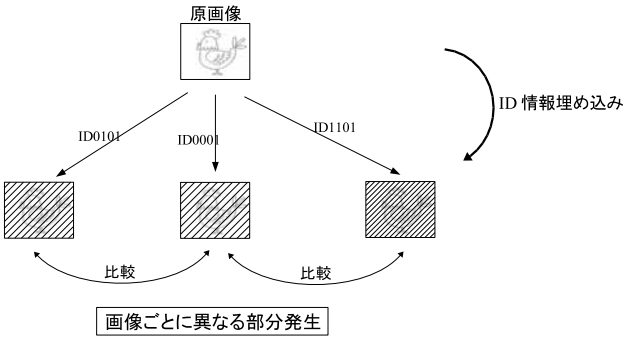

図 3: ID 情報の埋め込み

また動画においては，その動画を静止画像としてキャプチャしたものや，短い動画といったものには価値がないと考え，1 分以上の動画にのみ電子透かしが埋め込むことで十分であると考える．さらに，本システムでは前述のように 1 つのコンテンツに対して 2 種類の電子透かしを埋め込むが，これらの情報が個別に取り出せる必要がある．また，検出に関しては，透かしを埋め込む前のデータ (原版) を利用せずに検出を行うことができることが望ましい．

4. 実装

前章までの考察をもとに実際にインターネット有料放送システムを試作した．ここではその実装したシステムの仕組みについて述べる．

4.1 システムの全体構成

今回実装したシステムの概念図を図 4 に示す．図のようにコンテンツを提供するカメラ PC，放送と鍵の配布を行う放送用 PC，コンテンツを受信再生する受信用 PC から構成され，PC 上でソフトウェアのみを用いて放送・再生するシステムとする．それぞれ OS として，カメラ PC は Windows XP，放送用 PC は Linux，受信用 PC は Windows 各種を用いた．なお，今回の実装では電子

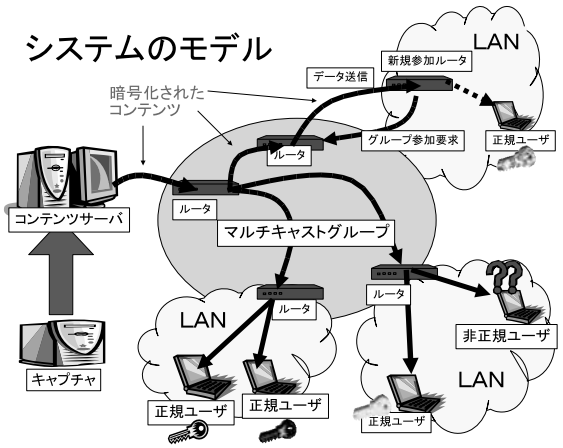

図 4: システムのモデル

透かしについては未実装である．運用実験としては京都大学内の LAN 環境をそのまま使用し，同一 VLAN で行った．

本システムではコンテンツとしてテレビ放送をそのまま利用する．カメラ PC ではテレビ放送をキャプチャして，後に述べる独自のプロトコル (VSUP) に従ってデータを送信する．放送用 PC ではカメラ PC から送られてきたデータをセッション鍵を用いて暗号化し，FEC による制御データを追加し，IP マルチキャストを用いて送信する．この際，セッション鍵は別途 Tracing Traitor 方式により暗号化してコンテンツと同様に送信する．受信用 PC ではまずあらかじめセッション鍵を得るための認証用鍵を放送用 PC より受信しておく．そして，暗号化されたコンテンツとセッション鍵を受信し，まず認証用鍵を用いてセッション鍵を取り出し，それを用いてコンテンツの復号・再生を行う．

4.2 カメラ PC の処理

カメラ PC はテレビ放送をキャプチャし，そのデータをサーバに送信する．その際，コンテンツを配信しようとしている側 (以後発信者と呼ぶ) からサーバに対しての要求や，サーバから発信者への制限などの情報の交換が必要となる．そこで，実装にあたりカメラ PC-放送用 PC 間の通信で用いるアプリケーションプロトコルとして VSUP(Video Streaming Uploading Protocol) を定義した．

4.2.1 VSUP

このプロトコルを作成するにあたってカメラ PC と放送用 PC 間の通信にとって必要な情報について整理する．まずサーバの立場から，発信者がコンテンツ配信を行う資格を持っているのかどうか確認する必要がある．次に，発信者が要求，指定したいこととしては，配信したいコンテンツの形式，使用したいチャンネル，そのコンテンツを視聴可能なユーザグループ，などといったものが挙げられる．具体的に，コンテンツの形式とは MPEG などといったデータ形式，ビットレート，フレームレートなどである．チャンネルとは，サーバが配信先として使用する IP アドレス，サーバが使用するポート番号などである．さらに，将来配信はユニキャストとマルチキャストどちらでも可能とすることを考えて，送信形式といったものも考えられる．このような要求に対してサーバは可能であるかどうか判断して応答する．

これまでの考えをふまえて，図 5，図 6 のようなコマンドと応答を定義した．

コマンド＜オプション＞…	内容
HELO<IP アドレス ><プロトコル ><バージョン >	通信の開始
USER< ユーザ名 >	ユーザ名の通知
PASS< パスワード >	パスワードの通知
TEST	TEST モードに入る
UPLOAD< データ形式 ><bitrate><framerate>	データ形式の指定
TESTSEND< バイト数 >	テストデータの送信
TESTEND	TEST モード終了
USERAUTH< グループ名 >	グループの指定
CHANNEL< 送信形式 ><IP アドレス ><ポート番号 >	チャネルの指定
SEND< フレーム番号 >< バイト数 >	データの送信
END	データの送信終了
PING	接続確認メッセージ
CONFIRM	放送が終了したか確認
QUIT	通信の終了

図 5: プロトコルのコマンド

コマンド	応答
HELO	100 OK
USER	110 OK
	120 BADNAME (ユーザ名が正しくない)
PASS	130 OK
	140 BADPASS(パスワードが正しくない)
TEST	200 OK
	210 NO
UPLOAD	220 OK
	230 NOAUTH (ユーザはこの方式で送る権利がない)
	240 OVERBIT (ビットレートが高すぎる)
	250 OVERFRAME (フレームレートが高すぎる)
	260 BADTYPE (対応していないデータ形式です)
TESTSEND	300 OK
	310 NO(バイト数が多すぎる)
	320 < 受信にかかった速度 >
TESTEND	330 OK
USERAUTH	400 OK
	410 NOT FOUND (グループ名が正しくない)
	420 BUSY (そのグループは使用中である)
CHANNEL	500 OK
	510 BADTYPE (送信形式が正しくない)
	520 BUSY (その IP アドレスは使用中)
	530 BUSY (そのポート番号は使用中)
SEND	600 < 次に待っているフレーム番号 >
END	610 OK
PING	630 PONG
CONFIRM	700 OK
	710 NOT YET

図 6: コマンドに対する応答

VSUP では TCP コネクションを確立してそのコネク

ション上で制御やデータ転送を行う．コネクションは発信者が自分の宣言をすることにより開始する．VSUP の機能を整理すると，ユーザ名とパスワードによる発信者についての認証，発信者によるコンテンツの形式，チャンネル，ユーザグループなどの放送形式の指定，実際の相互間の通信速度の測定，フレーム番号とバイト数を示してのコンテンツの転送などがある．VSUP では，実際の相互間の通信速度が，配信しようとしているコンテンツの形式ではリアルタイムに配信できないといったことがないよう，テストモードといった状態を置き，その状態で実際の通信速度を確認してからデータ通信に移る．さらに，カメラ PC はデータ送信終了後すぐにコネクションを切断せず，実際に放送が終了したのを確認してからコネクションを切断する．また，発信者からのデータ送信を一時中断する場合，コネクションは持続していることをサーバに知らせるために PING コマンドとして一定時間ごとに確認の通信をおこなう．

4.3 サーバの処理

サーバの処理としては大きく 2 つに分けることができる．コンテンツの再生に必要となるユーザ ID と認証用鍵を各ユーザへ配布することと，カメラ PC から送られてきたコンテンツをセッション鍵を用いて暗号化し，IP マルチキャストを用いて配信することである．また，信頼性を提供するため，サーバで FEC(Forward Error Correction) による制御データを付加する．

4.3.1 ユーザ ID と認証用鍵の配布

ユーザ ID と認証用鍵の配布は TCP を用いて独自のプロトコルで通信する．クライアントからユーザ名とパスワードを受信すれば，サーバはユーザ ID と認証用鍵を送信するといった簡単なものである．ここであらたにユーザ名といったものを使用するのは，ユーザ ID が非常に長く扱いづらいためユーザ ID と 1 対 1 に対応したユーザ名を用いることで取り扱いに関わる煩わしさを取り除こうとするものである．サーバは上のセッションが始まる前に，第 3 節のところで述べたように鍵の行列を乱数を用いて生成しておく．今回の実装では簡単のため，この鍵の行列の大きさを 1024 行 ×32 列とし，認証用鍵は各行から 1 つ取り出すことにより生成することにする．よって，この行列のサイズは 4096 バイト，ユーザ ID は $1024\times \log_2 32$=640 バイト，認証用鍵 128 バイトとなる．

4.3.2 コンテンツの暗号化・送信

サーバはカメラ PC から送られてきたデータストリームをパケット化し，そのパケットごとにセッション鍵で暗号化してからマルチキャストする．ここで，データパケット 1 つの大きさはヘッダを取り除いたデータ部分を 1024 バイトとする．暗号化の方法は今回の実装では簡単のため以下のようにする．まず適当なハッシュ関数を選択し，そのハッシュ関数に対して暗号化に用いるセッション鍵を入力とすることにより得られるハッシュ値と，カメラ PC から送られてきたデータの排他的論理和をとり，その結果を送信する．しかし，このままでは暗号強度が高くないため，将来は AES や DES などの標準暗号を用いて暗号化することを考えている．

送信に関しては IP マルチキャストを用いるが，後に述べる独自のヘッダを付加した後，上位プロトコルとして RTP[11](Real-Time Transport Protocol) を用いる．ヘッダの各部分の説明は省略するが，今回の実装ではペイロードタイプとシーケンス番号を主に使用する．ペイロードタイプはそのパケットに含まれるペイロードの種類が記述されており，これによりそのパケットが本システムの放送に用いるパケットであることを判断する．また，シーケンス番号には一連のパケットにおける順序番号が記述されており，これにより，パケットの欠落の検出などを行う．

独自ヘッダとしては，コンテンツ ID とパケットタイプを指定する．コンテンツ ID は 24 ビット，パケットタイプは 8 ビットとする．独自ヘッダのパケットタイプにより，コンテンツデータのパケット (以後データパケット) なのか，FEC による制御用のパケットなのか (以後制御パケット)，セッション鍵の行列のパケットなのか (以後セッション鍵パケット) ということを判断する．

ネットワーク上でのパケットの欠落としては単独の 1 つのパケットが欠落する場合と，バースト誤りによりいくつかまとまって欠落する場合が考えられるが，今回の実装ではその両方に対処し，かつ実装が容易ということから FEC として図 7 のような方式を用いる．図のように 100 個のデータパケットを 10 行 10 列に配置し，それぞれの行，列ごとに排他的論理和をとり，制御パケットとして送出する．これにより，連続する 10 個のデータパケットごとに 1 パケットまでの欠落を回復でき，連続する 100 個のデータパケットごとに 1 回のバースト誤りによる連続 10 個以内のパケットの欠落を回復できる．

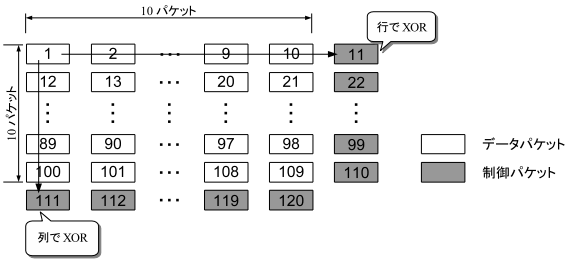

図 7: FEC の方法

4.3.3 セッション鍵の配布

サーバはコンテンツ配布に際し，セッションごとにセッション鍵を乱数を用いて生成する．セッション鍵のサイズは認証用鍵と同じ 128 バイトとする．生成されたセッション鍵と，サーバが認証用鍵をクライアントに配布する際に生成した鍵の行列とで排他的論理和をとることにより暗号化されたセッション鍵の行列を得る．今回の実

装ではこのセッション鍵の行列をセッション鍵パケットとしてデータパケットと制御パケット合わせて 120 パケット送信するごとに 1 回ずつの割合で繰り返し送信する．これはビットレート 1.5Mbps のコンテンツの場合約 0.1 秒に 1 回の割合である．これにより，セッションの途中で参加したユーザでも約 0.1 秒以内に受信を開始することができる．サーバの処理をまとめたものを図 8 に示す．

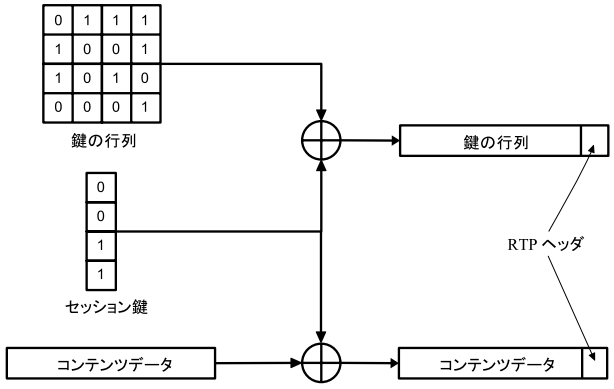

図 8: サーバの処理

4.4 クライアントの処理

クライアントの処理は，サーバから送られてくるデータを受信し，Tracing Traitor 方式によりセッション鍵を復号すること，そして，そのセッション鍵を用いて暗号化されたコンテンツを復号し，再生することである．

4.4.1 セッション鍵の復号

クライアントプログラムでは，ユーザ名とパスワードを設定しておくことにより，ボタン 1 つでユーザ ID と認証用鍵を取得することができる．クライアントはこの操作を課金単位ごとに行う必要がある．

セッションが始まると，まず受信データの独自ヘッダを調べることにより，セッション鍵パケットが届くのを待つ．このパケットは 0.1 秒以内に受信できるはずである．このパケットを受信したら，第 3 節で説明した Tracing Traitor 方式に従い，各行からユーザ ID に対応した箇所のデータを選択することにより，暗号化されたセッション鍵を得る．この暗号化されたセッション鍵に認証用鍵との排他的論理和をとることによりセッション鍵を復号する．

4.4.2 コンテンツの復号・再生

クライアントプログラムは前項で得たセッション鍵をサーバで用いたのと同じハッシュ関数の入力としてハッシュ値を得る．このハッシュ値をデータパケットから RTP ヘッダと独自ヘッダを取り除いたものと排他的論理和をとることによりコンテンツを復号し，そして再生する．

表 2: データサイズ

サーバの鍵の行列の大きさ：1024×32	
セッション鍵の行列	4096 バイト
セッション鍵	128 バイト
ユーザ ID	640 バイト
認証用鍵	128 バイト

サーバのところで述べたように，受信パケットには FEC のための制御パケットが含まれており，独自ヘッダのシーケンス番号を調べて，パケットの欠落などがあった場合にはそれを補いながら再生する．クライアントの処理をまとめたものを図 9 に示す．

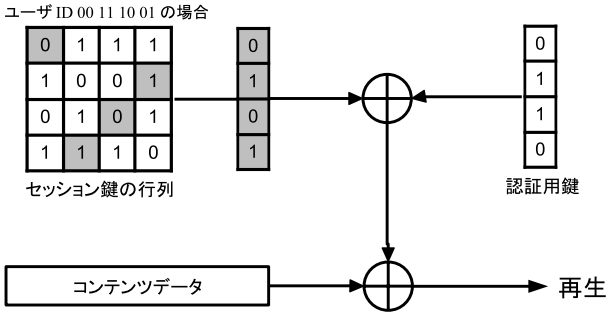

図 9: クライアントの処理

4.5 評価

ここでは実装したシステムに関して，Tracing Traitor 方式によるユーザ認証機構と，実運用における動作とについて評価する．

4.5.1 Tracing Traitor 方式による認証機構の評価

今回の実装では，認証用鍵配布のためにサーバが用意する鍵の行列の大きさを 1024 行 ×32 列としたため，3.1 項より，最大 65,536 人のユーザをサポートすることができ，4 人までが結託しても追跡することができる．また，セッション鍵の行列，セッション鍵，ユーザ ID，認証用鍵，それぞれの大きさは表 2 のようになる．現在の実装ではセッション鍵パケットはコンテンツデータのパケットを 120 パケット送信するごとに 1 つ送信する．1 パケットの大きさを 1024 バイトとすると，セッション鍵配布にかかるデータ転送量は全トラフィックの 3%程度に抑えられている．また，ユーザ ID と認証用鍵の配布にかかるデータ転送量はユーザは 1 人あたり 768 バイトであり，総データ転送量は最大で 50M バイト程度となる．現在一般に存在する通信速度が 10Mbps の web サーバで 1 分以内に 50M バイトのデータを送信可能なことや，ユーザ ID と認証用鍵の配布は時間的に分散されることからこの程度のデータ量であれば放送システムとして問題ないと考えられる．

4.5.2 実運用の評価

先に述べたシステムを実際に動作させ，

- カメラ PC-放送用 PC 間で VSUP を用いてテレビ放送データをアップロード
- 放送用 PC でそのデータを暗号化した後，FEC による制御パケットを付加し，独自ヘッダと RTP ヘッダを付けてマルチキャストにより送信
- マルチキャストされたデータを受信用 PC で受信し，データパケットを選択，復号することにより再生

といったことがリアルタイムに動作することが確認された．受信用 PC での実行画面を図 10 に示す．

今後はこれらに電子透かしの埋め込み処理を追加することを考えているが，そのような場合にもリアルタイムに処理することが可能かどうかという点が課題となる．

図 10: 受信用 PC での実行画面

5. 結論

スケーラビリティを確保することができ，不当な視聴や不正コピーを抑止することができる，有料インターネット映像放送システムを構築することを目指し，必要となる技術やその構築法について考察した．

ネットワークにおけるトラフィックの面，暗号の強度の面，各部分でかかる処理時間の面について検討した結果，配信には IP マルチキャストを用い，ユーザ認証には Tracing Traitor 方式を用い，不正コピー対策としてコンテンツに電子透かしを埋め込むことで，目標とするシステムが実現可能であることを示した．

この結果をふまえて，IP マルチキャストを用いて配信し，Tracing Traitor 方式のユーザ認証機構を用いた，有料インターネット映像放送システムを実装し，運用実験によって方式の実用性を確かめた．

今後の課題としては以下のようなことがある．今回は京都大学内の LAN 環境の同一 VLAN を利用して運用実験を行ったが，今後はより広域な実際のインターネット放送に近い環境での運用実験を行うことにより実用性を確かめたい．また，電子透かしについて具体的手法を提案し，実験により画質，処理速度，耐性などについて評価することや，有料放送システムとして実装することを行いたい．具体的には DirectShow のフィルタの 1 つとして組み込むことを考えている．

参考文献

[1] nullsoft SHOUTcast HomePage. http://www.shoutcast.com/

[2] PeerCast HomePage. http://www.peercast.org/

[3] Hugh Holbrook, Bradley Cain :Source-Specific Multicast for IP, Internet Draft (2003)

[4] 小野東 :電子透かしとコンテンツ保護 (オーム社, 2001)

[5] H. Harney, C. Muckenhirn :Group Key Management Protocol (GKMP) Specification, RFC2093 (1997)

[6] H. Harney, C. Muchenhirn :Group Key Management Protocol (GKMP) Architecture, RFC2094 (1997)

[7] A. Fiat, M. Naor :Broadcast Encryption, Proc. Advances in Cryptology-Crypt'93 (1994), pp. 480-491

[8] B. Chor, A. Fiat, M. Naor, B. Pinkas :Tracing Traitors, Lecture Notes in Computer Science, Vol. 839, pp. 257-270 (1994)

[9] 上原哲太郎, 川北良一, 辻義一, 佐藤敬, 山岡克式, 泉裕, 斎藤彰一, 國枝義敏, 結城皖曠 :IP マルチキャストを用いたユーザ認証つきインターネット放送システム, 情報処理学会論文誌 第 44 巻 第 3 号 (2003), pp. 610-624

[10] Deepa Kundur, Kannan Karthik :Video Fingerprinting and Encryption Principles for Digital Rights Management, Proceedings of the IEEE, Vol. 92, pp. 918-932 (2004)

[11] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson :RTP: A Transport Protocol for Real-Time Applications, RFC1889 (2000)