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

上原 哲太郎[†1] 川北 良一[†1] 辻 義一[†1]
佐藤 敬[†2] 山岡 克式[†3] 泉 裕[†4]
齋藤 彰一[†1] 國枝 義敏[†1] 結城 皖曠[†5]

インターネット上で多数の顧客に同時に有料放送サービスを実現するための機構について考察した．クライアント端末にユーザ識別のための特殊なハードウェアを用いることなく，また Internet のインフラストラクチャにも大きな変更を加えない条件の下で，IP マルチキャスト上でユーザ認証を行う仕組みについて比較検討した．ユーザ数のスケーラビリティを確保するためにはコンテンツは全ユーザに同じデータストリームとして到達するのが望ましく，また，安全性の確保のためには暗号化された単一データストリームを各正規ユーザ固有の異なる鍵で解読できるべきである．Naor らの提案による放送型暗号および Tracing Traitors 方式がこのようなシステムの実現に適していることを示した．また，この方式による放送システムの実装例として，IP マルチキャストを用いたユーザ認証付き音楽放送システム MusicCast/AS を実装した．サーバおよびクライアントは Linux 上のソフトウェアとして実装されており，ネットワークインフラストラクチャとしての学内キャンパスも特に変更なく実現できた．このシステムを用いて，音楽データを暗号化して放送する運用実験によって方式の実用性を確かめた．実装されたシステムのスケーラビリティおよび配布された鍵の安全性についても議論した．

An Internet Broadcasting System with User Authentication on IP Multicasting

Tetsutaro Uehara,[†1] Ryoichi Kawakita,[†1] Yoshikazu Tsuji,[†1]
Takashi Sato,[†2] Katsunori Yamaoka,[†3] Yutaka Izumi,[†4]
Shoichi Saito,[†1] Yoshitoshi Kunieda[†1] and Kiyohiro Yuki[†5]

This paper presents a methodology to realize a pay-broadcasting system which can supply the contents to support a number of clients simultaneously on the Internet. Various user-authentication methods on IP multicasting are compared under the constraint that the system should be constructed without any specific hardware for user identification, and it also must be installed without any changes on the network infrastructure of the Internet. To support scalable number of users, the contents of the system should be supplied to the users as the same data-stream. To keep the security of the contents, the encrypted single data-stream should be decoded by different keys individually distributed to the authorized users. Broadcasting Encryption and Tracing Traitors Schemes proposed by Naor et al. are considered to be suitable to realize such system. As a prototype of the system, a music broadcasting system named “MusicCast/AS” was implemented with user-authentication on IP multicasting. Its server and clients are implemented as software running on common Linux systems and the system do not need any changes in the existing campus network as a testbed. The practicality of the methodology was verified by the experimentation to broadcast encrypted music data to the campus network. The scalablity of the users and security of the keys are also discussed on the implementation.

1. はじめに

IT 革命とも呼ばれる劇的な技術の発達に従い，近年国内外のネットワークインフラストラクチャの整備

†1 和歌山大学システム工学部
Faculty of Systems Engineering, Wakayama University
†2 北九州市立大学国際環境工学部
Faculty of Environmental Engineering, The University of Kitakyushu
†3 東京工業大学学術国際情報センター
Global Scientific Information and Computing Center, Tokyo Institute of Technology
†4 和歌山大学システム情報学センター
Center for Information Science, Wakayama University
†5 メディア教育開発センター
National Institute of Multimedia Education

は急激に進展し，インターネットの高速化と広帯域化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可能になってきた．今後 FTTH（Fiber To The Home）などの高速接続サービスが一般家庭や多くの企業・団体に普及し，マルチメディア・ストリーミング通信が実用に達したことが多くのユーザにも実感されるようになれば，その需要は趣味やビジネスなど様々な面で，爆発的に増加すると予想される．特に，映像・音楽製作や放送に関わる企業などは，これらが保有する映像・音楽のコンテンツをインターネット上でストリーム放送すること，すなわちインターネット放送サービスの提供を新たなビジネスとして開始しようとしている．

現在，インターネットを使った映像や音声の放送は，RealNetworks 社の RealSystem [1]，Microsoft 社の Windows Media [2]，Apple 社の QuickTime [3]などを用いた，64 kbps から 256 kbps 程度までの比較的狭帯域向け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．ユニキャストによる実装は，ユーザが好みの時間タイミングで提供できる，すなわち，いわゆる「オン・デマンド」サービスが可能である．その反面，1 対 1 の通信であるために，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大する．よって原理的に，クライアント数に対するサービスのスケーラビリティを確保できず，たとえば数万人程度の大規模ユーザに対するコンテンツの同時提供は，現在のインターネットインフラストラクチャにおいては，サーバの冗長化・分散化なしには実現不可能であり，それはコンテンツ供給者に大きな負担を強いる結果となる．

上記の問題を解決するには，IP マルチキャスト [4] の利用が有効である☆．IP マルチキャスト通信では，あるマルチキャストアドレスに参加したホストすべてに対して同一のストリームデータを配送し，1 対多または多対多の通信を実現している．よってサーバの処理負荷やサーバからのデータ送信負荷が，クライアント数に依存せず一定にできるため，インターネット放送システムのような，不特定多数のクライアントを対象にいっせいにデータ配送するシステムに適している．現状の IPv4 における IP マルチキャストは実験段階であり，限られた範囲でしか運用されていない．しかし，IPv6 [6]において IP マルチキャスト [7]が標準機能として実装されていることや，マルチキャストの発信元の数を制限して配送やアドレス管理を容易にする Source Specific Multicast（SSM）[5]の提案があることから，今後は普及が進み，一般に利用が可能になると予想できる．

以上のことから，インターネットの広帯域化とマルチキャストの普及によって，インターネットによる動画像などの放送システムは近い将来実用段階を迎えると考えられる．これをビジネスとして成立させるために，コンテンツ供給者がユーザから個別に課金を徴収する有料放送サービスの要求が高まってくると考えられる．そこで本研究では，そのようなサービスの実現可能性について考察する．有料コンテンツ放送ビジネスを実現するには，放送の際に何らかの機構によってコンテンツの保護を行い，正規ユーザのみがコンテンツを視聴できるようにする必要がある．また，正規ユーザが故意に他の不正ユーザによる視聴を手助けすることがないように抑止することも必要である．本論文では，現在の IP マルチキャストを利用してシステムを構築でき，かつ，正規ユーザだけがコンテンツの視聴が可能である暗号システムについて検討し，暗号化システムの安全性とトラフィックに対する影響の双方から評価を行う．また，Pay-per-view システム，すなわちコンテンツごとに課金するシステムの実現例として試作した IP マルチキャスト向け音楽放送システム MusicCast/AS について述べる．

本論文の構成は下記のようになっている．2 章では，インターネット放送システムのモデルと要求仕様について述べる．3 章では，暗号化手法とユーザ認証機構を検証する．4 章では，今回プロトタイプとして実装した IP マルチキャスト向け音楽放送システム MusicCast/AS について述べる．5 章では，MusicCast/AS の評価を述べる．

2. 対象とする放送モデル

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目標とするサービスの形態と，前提にしたインフラストラクチャや技術について述べる．

2.1 放送システムのモデル

本研究が構築を目指すインターネット有料放送システムは，以下のようなサービスを行うものである．

2.1.1 配 規 規 模

本研究が目標とするシステムは，最大で数万人から数十万人程度のユーザが，同一のコンテンツを同時に視聴できるシステムである．同一コンテンツの有料放

☆ RealSystem, WindowsMedia, QuickTime など既存システムもマルチキャストを用いたサービスが行える機構は用意されているが，ユーザ認証機構が不十分である．

送であることであることからユーザ数は限界があると考え，数千万人から数億人の規模の同時視聴はここでは考慮しない．

この放送システムをユーザ数に対して十分にスケーラブルにするためには，放送時にサーバが送出するコンテンツデータ量がクライアント数に比例して増加することはないように工夫する必要がある．また，有料放送であることから，ユーザはコンテンツデータ受信時に何らかの認証を受けなくてはならない．この認証に必要なデータ通信量とユーザ数の相関も考慮する必要がある．これらの条件の下でコンテンツ配送と認証の方式を検討した．

2.1.2 コンテンツ

このシステムの対象となるコンテンツは，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．コンテンツの全データが全クライアントに届くことを必ずしも保証しなくてよい．クライアントには，データの欠落はコンテンツの品質劣化として観測されるが，欠落部分以外のデータは正しく受信・復号し再生できるものとする．データ欠落がコンテンツ再生の大きな妨げにならないよう保証する手法については別途議論する．

2.1.3 課 金 方 式

ユーザに対する課金方式は，2 通り考えられる．1 つはいわゆる Pay-per-view 方式や月単位契約方式のように,「料金と引き替えに，ある特定のコンテンツの視聴，あるいはある一定期間の視聴のための権利となる鍵を受け取る方式」である．この場合，一度ユーザに与えられた権利（鍵）は，サーバ側から無効にすることができず，またユーザも途中で権利を返却できない．その代わり，ユーザはある課金単位のコンテンツの送信の放送が開始された後であっても，料金を払えば途中から視聴が可能であるものとする．これはたとえば，月契約において，月の半ばからでも視聴が可能になる方式である．本論文ではこの方式を単に **Pay-per-view 方式**と呼ぶ．

もう 1 つの方式は，ユーザが任意の時点でシステムに参加したり脱退したりできる方式である．いい替えれば，サーバが任意の時点で特定のユーザの視聴を即座に禁止できる方式である．これはたとえば，秒単位のようなごく短い時間単位で課金するシステムや，月単位であっても契約途中でのユーザのキャンセル（課金の返金）に応じるシステムが考えられる．この方式を本論文では **Join-leave 方式**と呼ぶ．

これら両方式を比較すると，Join-leave 方式の方がユーザはコンテンツに対し必要な部分のみ最小限の課金で済ませられるため優れているといえる．しかし，コンテンツ提供者側にとっては特にユーザ数が多くなるとユーザ管理が複雑になり負荷が高まる．よって，これら両方式はアプリケーションによって使い分けられると考えられる．

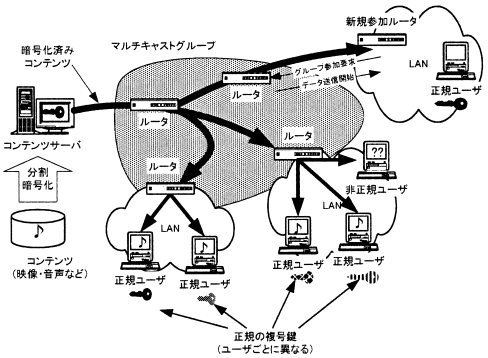


図 1 インターネット有料放送システム
Fig. 1 A pay-broadcasting system on the Internet.

これらいずれの方式においても，各ユーザには暗号化されたコンテンツを解読するための鍵が配布される．ここで仮にコンテンツを共有鍵で暗号化して配信する場合，全ユーザに同じ鍵が配布されるため，正規ユーザによって鍵が漏洩されれば，容易に不正ユーザもコンテンツの視聴が可能になってしまう．しかもこの場合，たとえ不正ユーザの存在が発覚しても，その不正ユーザに鍵を漏洩したユーザを追跡することは困難である．

そこで，ここでは，かならずユーザごとに内容が異なる鍵を配布する．この鍵を各ユーザの**認証用鍵**と呼ぶ．これにより，もし正規ユーザから不正ユーザに鍵が漏洩されても，不正が何らかの原因で発覚すれば不正に関わった正規ユーザを追跡することができるため，漏洩そのものを抑止できる．

このようなシステムの概念図を**図 1** に示す．以後，このような条件の下でサービスを行うために考慮すべき問題を議論する．

2.2 前提とするインフラストラクチャ

ここで考えるシステムでは，前述のとおりインターネットでの放送システムにユーザ数に対するスケーラビリティを持たせることが必要である．特にコンテンツのデータ送信においては，同時に同じデータグラムを多数のクライアントに伝えられる，IP マルチキャストを利用するべきである．

現在のマルチキャストの実装では，データの配送はマルチキャストグループ単位で管理されている．マルチキャストでのデータグラムの経路制御はサブネット

単位で行われており，サブネット内でのマルチキャストグループへの参加は，IPv4 では IGMP [8]を，IPv6 では MLD[15]を用いて管理されている．

この IGMP/MLD には現在ユーザ認証機構は存在せず，マルチキャストアドレスさえ分かっていれば，どのクライアントでも無条件でマルチキャストグループに参加できる．よって，有料放送においてマルチキャストを利用しようとすると，正規ユーザ以外にも容易にデータグラムの受信が可能になる．この問題は広く認識されており，IETF の The Group Security Research Group（GSEC）[20]などで討議，研究されている．また GSEC とは別に，IGMP にユーザ認証を加える拡張が提案されている[21]．しかしこれも標準として採用されるにはいたっておらず，現時点では実際には利用できない．また，仮に IGMP にユーザ認証機構があったとしても，末端の LAN 内ではマルチキャストデータは事実上ブロードキャストされるため，正規ユーザが視聴に使用している LAN に接続している他のクライアントは，容易に配送データを傍受できる．

そこで本研究では，利用できるインフラストラクチャは，通常の IGMP によって管理されたマルチキャスト環境を含む現在のインターネット環境をそのまま利用することとする．つまり，放送サーバから送信されたコンテンツのデータグラムが，マルチキャストによって何らかの形でクライアントに届けられる以外の仮定を置かない．よって，サーバからのデータ配送は，非正規ユーザの傍受に耐えるよう適切に暗号化されている必要がある．この暗号化の際の計算量やデータ量の増加を抑えるため，コンテンツは共通鍵を用いて暗号化することとする．このコンテンツを復号するための共通鍵を，以下では**セッション鍵**と呼ぶ．このセッション鍵を何らかの方法で正規ユーザだけに安全に配布することにより，有料放送システムを実現できる．

なお，このセッション鍵が正規ユーザから不正ユーザに漏洩された場合も不正ユーザに対する視聴が可能になるが，これは認証用鍵の入力からコンテンツの復号化までの機構をブラックボックス化したり，セッション鍵を頻繁に変化させたりするなどの手法で回避できると仮定して，本論文では議論しない．

2.3 システムの評価基準

ここでは，本研究で提案するシステムを設計するうえで，すでにあげた要求を満たすための必須条件となる項目とトレードオフとなる評価項目をあげる．

2.3.1 トラフィックとスケーラビリティ

システムをユーザ数に対してスケーラブルにするためには，放送サーバから送出される単位時間あたりのデータ量（トラフィック）は，ユーザ数に大きく影響を受けないことが望ましい．特に，コンテンツ本体のデータ転送量は，ユーザ数によらず一定であることを必須条件とする．本システムでは前述したとおり，コンテンツは単一のデータグラムとして，共通のセッション鍵で暗号化し，マルチキャストで全クライアントに配布することとしているため，トラフィックはユーザ数によらず一定となり，この条件を満たしている．

一方，各正規ユーザにセッション鍵を配布する際に必要となるデータ転送量は，各ユーザに異なる認証用鍵を配布しているため，ユーザ数とともに増加することは避けがたい．そこで本研究では，2.1.1 であげた配送規模に対応するため，ユーザ数 n に対してセッション鍵の配送に必要なデータ転送量が $O(\log n)$ 以下に抑えられる暗号化方式を必須条件とする．

また，各ユーザに配布する認証用鍵の配布にかかるデータ転送量もより小さい方がよい．システムの初期化直後は，各ユーザに一度は認証用鍵を配布しなくてはならない．これを認証用鍵の初期化と呼ぶ．この際に転送されるデータの総量は認証用鍵のサイズとユーザ数の積となるので，認証用鍵のサイズは，暗号化強度が十分である範囲でより小さくするべきである．認証用鍵の初期後は，各ユーザに配布した認証用鍵を課金単位ごとに無効にして再配布するか，あるいは一度ユーザに配布した認証用鍵を複数の課金単位にわたって再利用させるかによって必要となるデータ転送量が変わるので，これも評価項目となる．

2.3.2 不正に対する耐性

コンテンツの放送時には不正なユーザによる視聴を防ぐため，セッション鍵は少なくとも数百ビットの長さを持つことを必須条件とする．

また，認証用鍵は各ユーザによって異なるものが配られるため，正規ユーザから不正ユーザに認証用鍵が漏洩されることは抑止できると考えられる．しかし，正規ユーザが何人か**結託**することによって不正ユーザのための認証用鍵を新たに生成できる場合がある．このようなことは不可能になっている暗号化方式が望ましいが，このような結託が避けられない場合は，不正な認証用鍵の生成に必要な結託ユーザ数ができるだけ多いほうがよい．また，不正な認証用鍵が作成できた場合も，その認証用鍵から結託に関わった正規ユーザが同定または推定できることが望ましい（これを結託ユーザの**追跡**と呼ぶ）．このような評価項目に従って暗号化方式を議論する．

2.3.3 配送プロトコルの信頼性

本システムによる放送のためのインフラストラク

チャは，通常のマルチキャストによるデータグラム送信のみを前提としている．そのため，データグラムが伝送路で欠落する可能性があり，その対処が必要となる．

マルチキャストにおける配送の信頼性を上げるための研究（Reliable Multicast に関する研究）はこれまで多く行われてきており，本システムに適用可能なものも多いと思われる．これらの手法は，抜け落ちたデータの再送を要請する NACK ベースの手法と，誤り訂正符号を用いて解決する FEC ベースの手法に大別できる[22]．NACK ベースの手法は，本システムが目指すような多数の受信者へのマルチキャスト放送時には，伝送路に障害が発生すると多数のユーザから再送要求が発生し，輻輳の原因となりかねないため不適当である．よって FEC ベース，すなわち何らかの誤り訂正符号の技術を用いて，伝送路の信頼性を向上させるべきと考えられる．

3. 暗号化とユーザ認証

本研究では，2 章で述べたような評価基準に従っていくつかの鍵管理・配布方式について検討を行ってきた．ここでは，従来から提案されているセッション鍵管理方式について述べ，本論文で述べるシステムを実現する際に適した方法の検討を行う．

3.1 GKMP

インターネット上での 1 対多通信のためのセッション鍵配布プロトコルとしてすでに GKMP[23),24)]（Group Key Management Protocol）が提案されている．GKMP は，あるグループ内の 1 対多通信の暗号化に使用するべき鍵をグループメンバに分配するプロトコルである．

GKMP では，以下の 3 種類の鍵を用いる．

GTEK：Group Traffic Encrypting Key コンテンツを暗号化するための鍵（セッション鍵）で，ある時間単位ごとに変化させる．ある時間単位 n で利用する GTEK を GTEK(n) と表記する．

GKEK：Group Key Encrypting Key GTEK を時系列に沿って変化させるために，次の時間単位で用いる GTEK を送るための暗号化鍵（セッション鍵を配送する配送鍵）である．GTEK と組にして時間単位で変化させる．ある時間単位 n で利用する GKEK を GKEK(n) と表記する．

SKEK：Session Key Encrypting Key グループに参加してきたメンバに，その時点での GTEK，GKEK を送信するための鍵である．時間では変化しないがユーザごとに異なる．ユーザ番号 i のための SKEK を SKEK(i) と表記する．これが本方式における認証用鍵となる．

まず，準備として GKMP を利用するユーザ i は，あらかじめ Group Controller（GC）から各ユーザ固有の認証用鍵 SKEK(i) の配布を受ける（認証用鍵の初期化）．

各ユーザがコンテンツの視聴を開始する場合には，GC はまずセッション開始時点（すなわち時間 0）におけるセッション鍵 GTEK(0) およびセッション鍵配送鍵 GKEK(0) を生成し，各ユーザに対しそれぞれの SKEK を利用して個別に送付する（セッション鍵の初期化）．

その後，以下のような手順でコンテンツを放送する．

(1) 単位時間 n においては，サーバは，セッション鍵 GTEK(n) を用いてコンテンツを暗号化し，マルチキャストにより送信する．各ユーザは，受信した暗号化コンテンツを，すでに所有している GTEK(n) を利用して復号化し，コンテンツを視聴する．

(2) 次の時間単位 n+1 が近付いたら，GC は，GTEK(n+1) と GKEK(n+1) を生成する．そして，この GTEK(n+1) と GKEK(n+1) を，GKEK(n) で暗号化し，マルチキャストを利用して全ユーザに送信する．各ユーザは，その時点ですでに配布されている GKEK(n) を用いて受信したデータを復号化し，新しい GTEK(n+1) および GKEK(n+1) を入手する．

(3) 次の時間単位 n+1 になったら，サーバは新しい GTEK(n+1) を用いてコンテンツを暗号化，送信し，ユーザはそれを復号化，視聴する．以下同様に時間単位ごとに繰り返す．

(4) この間新しいユーザが視聴開始をしたときは，GC がそのユーザの SKEK を用いてその時間単位で有効な GKEK と GTEK を配送する．

GKMP は，認証用鍵，セッション鍵ともユーザ数に無関係に決定できるので，必要とされる暗号強度の範囲内で小さくおさえることができる．よって，認証用鍵の初期化に必要なトラフィックは小さい．しかし，セッション鍵の初期化が必要となるため，セッション開始時のデータ転送量がユーザ数 n に対して $O(n)$ になり，2.3.1 項で設定した制限を超えてしまう．

さらに大きな問題として，GKMP は，セッションに参加しているユーザが単調に増加している間は問題なく機能するが，ある時点で課金切れを起こしたユーザがセッションから脱退する場合に，そのユーザを新しい課金時間単位において確実に排除することが難

しい．なぜならそのユーザはすでに新しい課金時間単位向けの GTEK/GKEK の配送を受けられる鍵を保持しているからである．このようなユーザを排除するためには，セッション開始時と同様に新たな GTEK および GKEK を全ユーザに個別に配布する必要があるが，このためにはセッション鍵の初期化と同じ手順によって行わなければならず，ユーザ数が多い場合には大量のデータ転送が必要になる．これは特に課金に Join-leave 方式をとる場合に問題となる．また，正規のユーザであっても放送データの受信を中断してから再開すると，その時点の GTEK/GKEK を GC に問い合わせなくてはならない問題がある．たとえば Pay-per-view 方式において課金時間の単位が月単位のように長時間である場合は，ユーザが課金時間内で受信の中断と再開を繰り返すと考えられる．この場合 GTEK/GKEK の GC への問合せがデータ転送量の増加を招き，ユーザ数に対するスケーラビリティが低下する．

3.2 階層型鍵配送サーバ

セッション鍵の配送にスケーラビリティを持たせるため，多数の鍵配送サーバをネットワーク上に木構造に配置して鍵の配送を階層的に行う手法がいくつか提案されている[25),26)]．この方法は大きなスケーラビリティと比較的強固なセキュリティが得られるが，現在のインターネット上の各所に鍵配送サーバを配置する必要があり，インフラストラクチャに大幅な変更を加えることになる．よってこの方式は 2 章で述べたような，ネットワークインフラストラクチャに大幅な改変を加えないという条件を満たすことができないため，検討から除外する．

3.3 放送型暗号

同一の暗号化データをマルチキャストにより各ユーザに送信し，受信したそれぞれのユーザはあらかじめ配布されている固有の異なる鍵により復号を行うことができる，放送型暗号[27)]が Fiat らによって提案されている．

放送型暗号では，それぞれのユーザに対して鍵の集合を配布する．たとえば，最大 n 人のユーザをサポートする際には，n 個以上の鍵，たとえば $\{K_1, K_2, \cdots K_n, K_{n+1}\}$ を用意する．そして，各ユーザに対しユーザ P_1 には鍵の集合から K_1 を除いたもの，ユーザ P_2 には K_2 を除いたもの，というようにそれぞれ一部の鍵を除いた残りの鍵の集合を配布し，これを認証用鍵とする（**図 2**）．そしてデータ送信時には，そのデータを受信させたくないユーザが持っていない鍵を集めて合成し，それをセッション鍵として暗号化する．たとえば図 2 においてサーバが P_1，P_2 だけにコンテンツを送信する場合には，K_3 から K_n までの鍵を合成して暗号化を行い，合成に使用した鍵番号と暗号化したコンテンツを同時に全ユーザにマルチキャストすればよい．P_3 から P_n までのユーザにとっては，データは受信できてもセッション鍵の合成に使われた鍵集合のいずれかの部分が欠落しているため，正しく復号することができない．

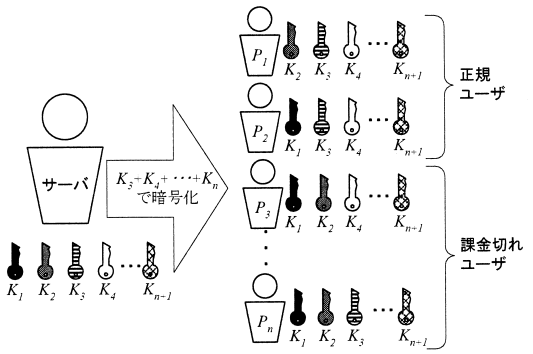


図 2 放送型暗号
Fig. 2 Broadcast encryption.

現実には，このとおりの実装を行うと各ユーザがあらかじめ保有している認証用鍵のサイズが，システムのサポートする総ユーザ数 n に比例し，認証用鍵の初期化に必要なデータ量を増大させる．しかし，認証用鍵の基となる鍵の集合の生成に一方向性関数による疑似乱数を用いることによって，各ユーザの認証用鍵のサイズを $O(\log n)$ にまで小さくすることができることが文献 27) に示されている．また，本方式はセッション鍵は暗号化されたうえで同一のものが全ユーザに届けられるので，そのデータ量はセッション鍵の暗号化後のサイズそのものとなる．これは文献 27) によるとユーザ数 n に対し $O(\log n)$ で済む．

この方式の最も大きな短所は，ユーザ同士の結託による不正受信の可能性があることである．たとえば図 2 の例では，任意の 2 名のユーザが結託し，それぞれが持っている鍵を互いに交換すると，$\{K_1, K_2, \cdots, K_n, K_{n+1}\}$ すべてが揃うため，その後結託したユーザはつねにデータの復号化が可能になってしまう．これを避けるため，鍵を最大ユーザ数に対して十分な数用意し，各ユーザに配布する鍵の個数をすべての鍵の個数に対して相対的に小さくすることにより，すべての鍵を揃えるために必要なユーザの結託人数を増加させる．文献 27) によると，最大ユーザ数を n とし，ユーザのうち k 人までが結託しても不正に復号できなくするためには，各ユー

ザの保有鍵サイズを $O(k\log^2 n)$ とする必要があり，また一度のセッション鍵配布に必要なサーバからのデータ転送量は $O(k^3\log n)$ となる（ストレージ優先符号化と呼ぶ）．さらに，ユーザの保有鍵サイズを $O(k\log k\log^2 n)$ とすることにより，サーバからのデータ転送量を $O(k^2\log^2 k\log n)$ まで減少させることができる（トラフィック優先符号化と呼ぶ）．

いずれにせよ，放送型暗号では各ユーザが持つべき認証用鍵のサイズが他の方式に比べると大きく，全ユーザに認証用鍵を頻繁に再配布する負担は大きい．また結託に対する強度を上げていくとセッション鍵の配布に必要なトラフィックも急速に増大する．しかし，本方式はユーザが認証用鍵を保有したままでもサーバ側から能動的に視聴を抑止できるため，課金方式として Pay-per-view 方式はもちろん Join-leave 方式にも適用可能な点が優れている．

3.4 Tracing Traitors Open One-Level Scheme（TT-OOLS 方式）

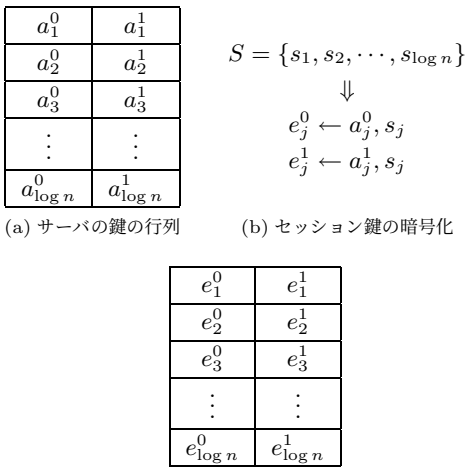
Chor らによる Tracing Traitors 方式[28]は，放送型暗号と同様にサーバで鍵の集合を用意し，各ユーザに部分集合を配布する方式である．この方式にはいくつかバリエーションがあるが，基本的な方式である Open One-Level Scheme を TT-OOLS 方式と呼ぶことにする．この中で最も単純な例として，n 人までのユーザをサポートする場合を以下に簡単に説明する．

(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 $a_1^0, a_1^1, a_2^0, a_2^1, \cdots, a_{\log n}^0, a_{\log n}^1$ を作成する．これを**図 3** のような行列で表す．

(2) 各ユーザに ID 番号を付与し，それに基づいてこの $2\log_2 n$ 個の鍵のうち $\log_2 n$ 個をそれぞれのユーザの認証用鍵として配布する．ユーザ ID が i であった場合，i は $\log_2 n$ 桁の 2 進数で表せる．この 2 進数の下位から i 番目の桁が 0 なら a_i^0 を，1 なら a_i^1 を付与する．たとえば，$n = 16$ の場合においてユーザ ID が 6 ならば，そのユーザは a_1^0，a_2^1，a_3^1，a_4^0 の 4 種類の鍵を認証用鍵として得る．結果として，ユーザ ID が異なれば保持している認証用鍵は異なる．ここまでの処理は，セッション開始までにあらかじめ行っておく．

(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をユーザに送信する．このときのセッション鍵を S とすると，サーバはこの S を $\log_2 n$ 個の鍵 $s_1, s_2, \cdots, s_{\log n}$ に分割する．次に s_j を a_j^0，a_j^1 それぞれを用いて暗号化し，e_j^0，e_j^1 を得る．この暗号化された $2\log_2 n$ 個のセッション鍵 e で図 3 の (c) のような行列を作成し，全ユーザに送信する．

(4) 受信した各ユーザでは，暗号化されたセッション鍵の行列を受け取り，その中からユーザ ID に相当する要素を取り出して，保持している $\log_2 n$ 個の認証用鍵で復号し，結合してセッション鍵 S を得る．たとえば前述のユーザ ID が 6 のユーザは，a_1^0，a_2^1，a_3^1，a_4^0 を持っているので，受信したセッション鍵の行列から e_1^0，e_2^1，e_3^1，e_4^0 を取り出し，s_1，s_2，s_3，s_4 を得る．その結果を結合することにより，セッション鍵 S を得ることができる．



(a) サーバの鍵の行列　(b) セッション鍵の暗号化

(c) セッション鍵の行列

図 3 TT-OOLS 方式における鍵の行列
Fig. 3 A key matrix for TT-OOLS method.

この方式によれば，n 人のユーザに対してセッション鍵を配布する際には，必要となる認証用鍵のサイズは，$O(\log_2 n)$ でしか増加せず，セッション鍵の放送にかかるトラフィックも $O(1)$（この場合 2 倍）でしか増加しないため，トラフィックの増加を低く抑えられる．

しかしこの TT-OOLS 方式も，正規ユーザ同士の結託があった場合，サーバの持つ元の鍵の行列が判明してしまう場合がある．たとえばここにあげた例ではユーザ ID が互いに排他的論理和になっている 2 ユーザが結託するとサーバの持つ鍵の行列が完全に判明するため，どのようなユーザ ID に対しても認証用鍵を不正に生成できる．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不正な認証用鍵が発行できる．

そこで，鍵の行列の行と列を広げ，不正な視聴に使

用された認証用鍵から結託に関わったユーザを追跡できるようにする手法が文献 28) に示されている．不正な認証用鍵は，結託に参加したそれぞれの正規ユーザの認証用鍵から鍵の行列の要素を部分的に取り出して連結したものである．よって，不正な認証用鍵のそれぞれの要素を持っているユーザの集合を得てその積集合をとっていくことで，結託に参加したユーザの推定が可能になる．詳細はここでは述べないが，結論としては，ユーザ数 n に対して k 人までが結託しても追跡できるようにするためには，サーバの鍵の行列を $4k^2 \log n$ 行 $2k^2$ 列とし，この行列とユーザ ID とから認証用鍵を発行する際に適切なハッシュ関数を適用すればよい．この際，各ユーザが持つ認証用鍵のサイズは $O(k^2 \log n)$ であり，またセッション鍵の配布に必要なデータ転送量は $O(k^4 \log n)$ である．

また，TT-OOLS 方式では複数のユーザが認証用鍵を部分的に共有しているため，あるユーザがセッションの途中で脱退した場合，そのユーザが保有していた認証用鍵をサーバ側から無効にすると，その認証用鍵と重複部分を持つ他のユーザの認証用鍵も無効になってしまう．よって，セッションの途中で特定のユーザの認証用鍵のみ無効にすることは困難である．つまり TT-OOLS 方式も GKMP と同様，課金方式として Join-leave 方式には向かず，Pay-per-view 方式にのみ適用可能といえる．

3.5 セッション鍵配布方式の比較

2 章で述べてきたような評価基準に従って，本研究では以上のようにセッション鍵の管理および配布方式について検討した．GKMP，放送型暗号，TT-OOLS 方式を比較した結果，**表 1** のようになった．

スケーラビリティについては，放送型暗号と TT-OOLS 方式はセッション鍵の初期化も更新もマルチキャストのみで行えるためユーザ数に対しスケーラビリティを持つといえるが，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り，スケーラビリティに欠ける．

各ユーザが持つ認証用鍵のサイズについては，ユーザ数 n に対して GKMP は $O(1)$，TT-OOLS 方式が $O(\log n)$ と小さいが放送型暗号は $O(\log^2 n)$ であり大きい．

セッション鍵の配送時に必要なデータ転送量は，ユーザ数 n に対して GKMP は $O(1)$ であるが，放送型暗号と TT-OOLS 方式は $O(\log n)$ と大きい．さらに，TT-OOLS 方式と放送型暗号には，ユーザの結託の問題がある．ユーザの k 人の結託への耐性を持たせようとしたときには，セッション鍵の配送時に必要なデータ転送量は，放送型暗号はトラフィック優先符号化の場合で $O(k^2 \log^2 k \log n)$，TT-OOLS 方式では $O(k^4 \log n)$ である．

表 1 セッション鍵配布方式の比較
Table 1 Comparison table for session-key distribution methods.

方式	GKMP	放送型	TT-OOLS
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
ユーザ結託への耐性	強固	弱	弱
Join-Leave 方式	不適	適	不適
Pay-per-view 方式	適	適	適

課金方式に関して Join-leave 方式を採用しようとすると，放送型暗号が最も本システムの実現に適している．ただし，認証用鍵の配布に必要なデータ転送量が大きいため，新規のユーザが頻繁に加入するようなアプリケーションでは放送型暗号でも実現が困難になる．他の 2 方式は，セッション途中で脱退したユーザが発生すると，GKMP の場合はセッション鍵の初期化，TT-OOLS 方式においては認証用鍵の再配布が必要になり，いずれも $O(n)$ 以上のデータ転送を必要とするため，Join-leave 方式に適用するのは困難である．

一方，課金単位内でのユーザの脱退を許さない Pay-per-view 方式については，3 方式とも適用可能である．ただしそれぞれ異なる性質を持つ．GKMP については，各ユーザが持つ認証用鍵は小さく，一度配布すれば再配布の必要はないが，セッションに参加するユーザには個別にセッション鍵の初期化が必要である．よって，課金単位が比較的長時間にわたり，各ユーザがこの間視聴と中断を繰り返すような運用には適していない．TT-OOLS 方式は，課金単位が切り替わった際に以前の課金単位に利用していたユーザが 1 人でも脱退していると認証用鍵を再配布する必要があるが，一度認証用鍵を配布すれば課金単位内では再配布の必要がない．放送型暗号は認証用鍵の再配布も原則として必要はなく，複数の課金単位にわたって認証用鍵の再利用が可能である．ただし Join-leave 方式の場合と同様に，新規のユーザが頻繁に加入してくる状況では認証用鍵の配布コストのため不利になる．

放送型暗号と TT-OOLS 方式についてはユーザの結託による不正視聴の可能性があるため，これについても評価する．

放送型暗号においては，ユーザの結託によりサーバが持つ鍵の集合すべてが分かるため，結託したユーザは，それ以降の課金単位においても，課金を納めずに引き続き視聴を継続できる．つまり，結託したユーザ

自身が利益を受けるため，ユーザが結託に参加する動機づけになりやすい．また．結託に参加したユーザが互いに秘密を守っている限りは，放送者側が結託の事実を察知する機会がない．さらに，結託したユーザが使用している鍵を第三者に配布した場合，放送者側がその不正な鍵を入手しても，結託に参加したユーザを同定することはできない．しかも放送型暗号では，不正状態を解消するにはサーバにおいて鍵の集合を再生成して認証用鍵を全ユーザに再配布する必要があり，これに必要なデータ転送量が大きい．

一方，TT-OOLS 方式においては，ユーザの結託によりサーバが持つ鍵の行列の全部ないし一部が判明しても，結託に参加したユーザ自身はそのままでは利益を受けない．利益を受けるのは，結託したユーザが新たに生成した不正な認証用鍵の配布を受ける第三者である．また，TT-OOLS 方式においては課金単位ごとに認証用鍵を再配布するため，課金単位が切り替わるたびにユーザが何度も結託して不正な認証用鍵を生成する必要がある．さらに，結託に参加した人数が鍵の行列生成時に設定した値 k 以下の場合には，第三者に引き渡された不正な認証用鍵から，結託に参加したユーザが追跡できる．ただし，セッション鍵配布のために必要なデータ量が，追跡可能になる結託人数 k に対し $O(k^4)$ にもなるため，結託人数に対する耐性をあまり強固にできない．

このような比較検討の結果は，以下のようにまとめられる．

- 課金システムに Join-leave 方式をとる場合には，無条件に放送型暗号を使用するべきである．
- 課金システムに Pay-per-view 方式をとる場合，ユーザの入れ替わりが激しい場合は TT-OOLS 方式が有利である．そうでない場合は放送型暗号が適している．
- GKMP は，比較的課金単位が短時間で，その課金単位内にセッションに参加したユーザが視聴を中断・再開しない場合は，セッション鍵の更新コストが非常に小さいので有利である．
- GKMP は結託攻撃とは無関係である．放送型暗号は TT-OOLS 方式より結託に対する耐性を強めやすい．しかし放送型暗号は結託に参加したユーザ本人が利益を受け，結託に成功すると追跡も難しいため，より結託攻撃に狙われやすい可能性がある．

以上より，インターネット有料放送システムの実現には，上記の結果をふまえた，構築しようとするアプリケーションに適した認証用鍵およびセッション鍵管理方式を採用するべきと考えられる．

4. Musiccast/AS の実装

これまでの検討結果をふまえて，インターネット有料放送システムの実現可能性を検証するため，Pay-per-view 型音楽放送システム MusicCast/AS を開発した．以下では，この実装について述べる．

4.1 システムの全体構成

MusicCast/AS は，放送・鍵配布サーバと視聴用クライアントからなるシステムである．実装にあたっては，計算機には IBMPC 互換機を，OS には Linux を用いた．運用実験に際しては，Gigabit Ethernet と Fast Ethernet で構成された和歌山大学内の LAN 環境をそのまま使用した．

課金システムは Pay-per-view 方式とし，課金単位は月単位など長期間を想定したたことから，暗号化には TT-OOLS 方式を採用した．

MusicCast/AS では，MP3 形式[16)~18)]でエンコードされた音楽データをコンテンツとして使用する．放送用サーバは，ストレージにある MP3 ファイルを分割，暗号化し IP マルチキャストを用いてネットワークに送出する．この際，コンテンツを復号するのためのセッション鍵は別途 TT-OOLS 方式を使って暗号化し，コンテンツと同様にネットワークに送出する．

クライアントプログラムは，暗号化・断片化されたコンテンツを受信して復号，結合して MP3 プレーヤに送り込む．この復号に必要となるセッション鍵は，コンテンツと同様に受信できるので，これを TT-OOLS 方式で復号化し使用する．

この TT-OOLS 方式での暗号化と復号化に必要な鍵は，放送サーバに構築した認証サーバで生成される．認証サーバでは，生成した鍵の行列を放送サーバに伝えるとともに，ユーザからの要請に応じてユーザ ID と認証用鍵を発行する．各ユーザがこのユーザ ID と認証用鍵を自マシンのクライアントプログラムに入力することにより，視聴が可能となる．

4.2 MusicCast/AS サーバの処理

MusicCast/AS サーバは送信サーバと，Web アプリケーションとして開発されているユーザ認証用鍵配布サーバとからなる．

4.2.1 ユーザ ID と認証用鍵の配布

まず鍵配布サーバでは，図 3 (a) で示したように，放送サーバが持つべき鍵の行列を乱数で生成し，保持する．今回の実装では，行列の行数，列数は可変だが，簡単のため鍵の行列の各要素は 1 ビット値とした．よって行数を r，列数を c とすると，セッション鍵の長さ

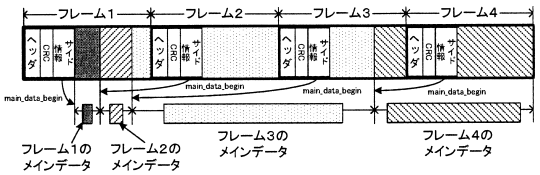


図 4 MP3 のフレーム構造
Fig. 4 A frame of MP3 file.

は r ビットであり，ユーザ ID のビット数は $r \times \log_2 c$ である．

4.2.2 MP3 ファイルの送信と暗号化

MusicCast 放送サーバでは，送信コンテンツを含む MP3 ファイルを暗号化して IP マルチキャストで送出する役割を持つ．

MP3 ファイルは，フレームと呼ばれる基本単位の集合体である[19]．フレームは**図 4** のような構造であり，各フレームのサイズはデータ形式とサンプリングレートなどから求まるある一定値（96〜1,441 バイト）である．フレームはヘッダ，サイド情報およびメインデータからなり，メインデータが実際の圧縮された音声情報を保持している．図 4 にも示されているとおり，メインデータは必ずしも固定サイズであるフレームに収まるとは限らないので，サイド情報が持つポインタに従って前後のフレームにまたがって存在する．よって，フレームが伝送路で 1 つでも欠落した場合，再生できるようにデータを整合させるまで数フレームを要する．そこで今回の実装では，簡単な誤り訂正符号を用いてフレームが欠落する可能性を減らしている．

放送サーバでの MP3 ファイルの分解・暗号化と送出の手順は以下のとおりである．まず MP3 ファイルをフレームに分解し，ヘッダ情報を解析する．これは特に各フレームのビットレートを得て，パケット送出間隔を調整するために必要である．このフレームを，1 つまたは数個まとめて最低 600 バイト以上のサイズになるように調整し，必要に応じてこれを暗号化する．その後に独自のヘッダをつけた UDP パケットとしてある特定の IP マルチキャストアドレスに送出する．ここで加える独自ヘッダには以下の情報を保持している．

- セッション ID（32 bit 符号なし整数）
- パケットの種類（非暗号化コンテンツ，暗号化コンテンツ，セッション鍵，パリティパケットの 4 種類）
- パケット番号（32 bit 符号なし整数）
- パケットのサイズ

今回の実装ではパケットごとに暗号化されているか否かのタグをつけてあり，コンテンツの暗号化と非暗号化を切り替えられるようにしている．暗号化は，現実装では簡単のため，セッション鍵の値から得られるハッシュ値と，コンテンツデータの内容との排他的論理和とした．MP3 のメインデータは圧縮されているためエントロピーが高く，このようなアルゴリズムでもセッション鍵の推測は抑止可能である．

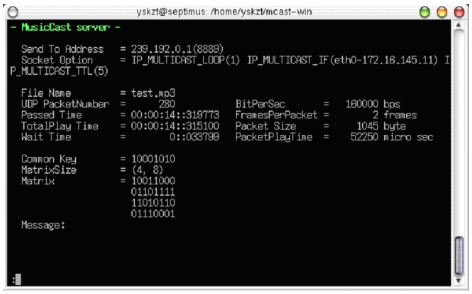


図 5 MusicCast/AS 放送サーバの実行画面
Fig. 5 A screen shot of MusicCast/AS server.

また，パケットの欠落に対処するため，コンテンツを含むパケットを 20 個送出するごとに，それら 20 個のパケットのうちセッション ID とパケットの種類を除いた部分の排他的論理和をとり，パリティパケットとして送出している．これにより，連続した 20 個のパケットごとに 1 パケットまでの欠落を回復できる．

なお，コンテンツデータ送出に際しては，各フレームから得られたビットレートにあわせ，パケットの送信間隔を調整している．

4.2.3 セッション鍵の配布

MusicCast/AS 放送サーバは，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．生成されたセッション鍵と，鍵配布サーバが持つ鍵行列とから TT-OOLS 方式によって暗号化されたセッション鍵の行列を得る．これをセッション鍵パケットとして，コンテンツと同様に独自ヘッダおよび行列の行数，列数を加えた情報として送信する．今回の実装では，セッション鍵パケットの送信はセッション開始直後だけでなく，放送中は 100 パケットに 1 回ずつの割合で繰り返して行っている．これはビットレートが 128 kbps のコンテンツの場合約 5 秒に 1 回の割合である．これにより，セッションに途中参加したユーザや，最初のセッション鍵パケットの受信に失敗したクライアントでも約 5 秒以内に受信が開始できる．

実際のプログラムの実行画面を**図 5** に，また放送サーバの処理の流れを**図 6** に示す．

4.3 MusicCast/AS クライアントの処理

MusicCast/AS のクライアントプログラムのおもな

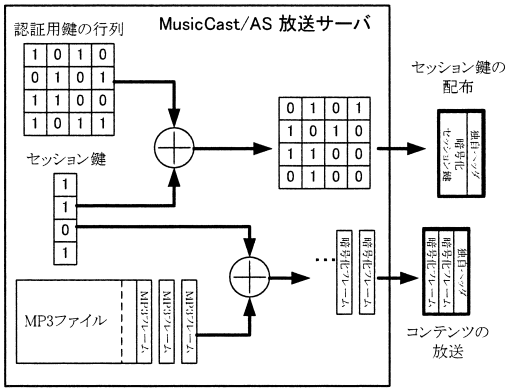


図 6 MusicCast/AS 放送サーバ内の処理
Fig. 6 Process flow in MusicCast/AS server.

処理には，セッション鍵の取得，コンテンツの復号化と結合，再生処理がある．

4.3.1 セッション鍵の復号

このシステムに参加するユーザは，ユーザ ID と，認証用鍵をあらかじめ受け取る必要がある．これらを得るための鍵配布サーバは Web アプリケーションとして実装している．**図 7** に，ユーザ登録と ID，認証用鍵の割当てを受ける Web サーバを示す．各ユーザはここでの操作によりユーザ ID とパスワード（すなわち認証用鍵）を得て，クライアントプログラムに手動で入力する☆．

認証用鍵の入力を行ったクライアントを起動して視聴を開始すると，クライアントプログラムは IP マルチキャストにより放送サーバから送られてくるパケットからセッション鍵パケットが送られてくるまで待つ．セッション鍵パケットは，4.2.3 項で述べたように，最悪でも 100 パケット以内に受信することができる．クライアントは，受信した暗号化セッション鍵の行列とユーザ ID，認証用鍵とから TT-OOLS 方式に従ってセッション鍵を取得する．

4.3.2 コンテンツの受信処理

セッション鍵が取得されれば，後は再生処理に進む．サーバからマルチキャストで送信されてくるパケットからヘッダを抜き取り，暗号化されている場合はセッション鍵を用いて復号化し，MP3 フレームを取り出して結合する．この際，ヘッダのパケット番号に欠落が見られた場合は，パリティパケットの到着を待ち，これを用いて復元を試みる．復元に失敗した場合は，ダミーのデータを含む仮のフレームを挿入してから結合する．

☆ 現在はこの操作は課金単位が切り替わるごとに必要であり煩雑となるので，将来的にはクライアントプログラムが直接鍵配布サーバと通信するように拡張する予定である．

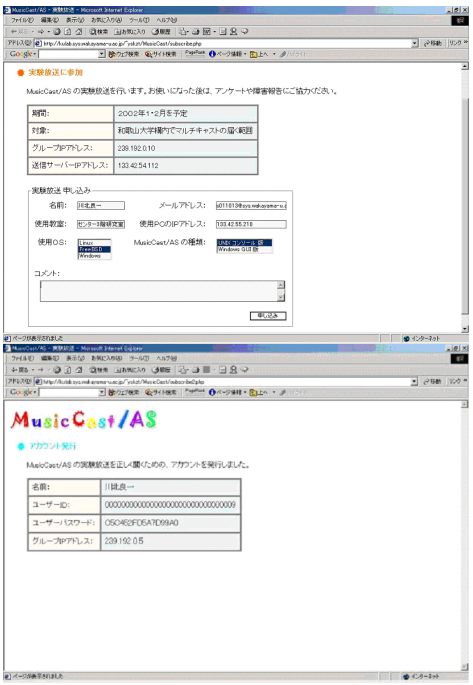


図 7 MusicCast/AS 登録サーバ
Fig. 7 MusicCast/AS registration page.

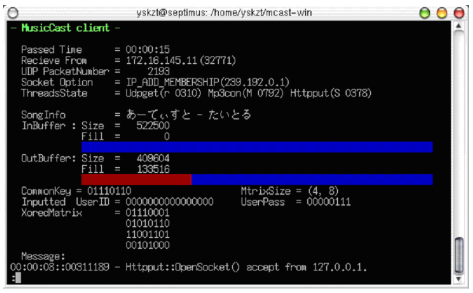


図 8 MusicCast/AS クライアントの実行画面
Fig. 8 A screen shot of MusicCast/AS client.

このようにして結合されたフレーム（すなわち MP3 ファイル）は結合が終わった部分からバッファを介して MP3 プレーヤに送られる．現在の実装では，クライアントプログラムを自ホストからのみ接続可能な HTTP サーバとして動作させ，一般のソフトウェア MP3 プレーヤに HTTP 経由でアクセスさせることによって実現している．クライアントの実行の様子を**図 8** に，処理の流れを**図 9** に示す．

なお，放送の途中でクライアントプログラムを起動した場合は，MP3 ファイルの中途のフレームからの再生となる．各 MP3 フレームはサイド情報にそれ以前のフレームのメインデータへのポインタを含む場合

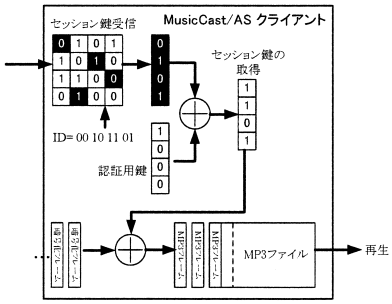


図 9 MusicCast/AS クライアント内の処理
Fig. 9 Process flow in MusicCast/AS client.

があるため，MP3 ファイルの中途からの再生はデータに矛盾が発生する．しかし，今回使用した MP3 プレーヤ☆では，MP3 フレームさえ正常であれば再生が可能になったフレームまで読み飛ばした後再生できることを確認した．

これに対し，認証用鍵の入力などを誤って入力してしまった場合など，正常なセッション鍵が復元できなかった場合は，MP3 ファイルも正しく復元できず，MP3 プレーヤによって再生できないか，ノイズの入った雑音として再生されることも確認した．

5. 評　　価

本章では，5 章にて述べた MusicCast/AS についての評価を行う．

5.1 TT-OOLS 方式による認証機構の評価

5.1.1 トラフィックへの影響

TT-OOLS 方式においては，認証のため，各ユーザに対して認証用鍵を行列の形で冗長に暗号化してから配布する．現在の実装では，放送時におけるセッション鍵の配布は 100 パケットにつき 1 パケットの割合で行われる．コンテンツデータのパケットは，CD 音質のサンプリングレートで毎秒 128 kbit のビットレートの場合約 850 バイトである．TT-OOLS 方式における暗号化されたセッション鍵のデータサイズは，たとえば最大 65,536 人のユーザをサポートし，4 人までの結託が起きてもユーザを追跡できるようにした場合，鍵の行列は 1,024 行 32 列となり，暗号化後のセッション鍵の行列サイズは 4,096 バイトである．この MP3 コンテンツを本システムで放送した場合，フレーム長セッション鍵の更新にかかるデータ転送量は全トラフィックの 4.5%程度である．またこのときの各ユーザの認証用鍵の大きさは 1,024 ビットすなわち 128 バイトと小さく，ユーザ数が最大の場合の総データ量も 8 メガバイトにすぎない．現在は認証用鍵の配布サーバの実装が WWW サーバになっているため，たとえば数分で数万のクライアントに対して認証用鍵を提供するのは困難であるが，専用の鍵配布サーバを実現すれば，数分間のうちに全クライアントに認証用鍵を配送することは十分可能であると考えられる．

5.1.2 TT-OOLS 方式のセキュリティ

すでに述べたとおり，TT-OOLS 方式は，認証用鍵を持った正規のユーザ同士が結託して不正な認証用鍵を新たに生成できる場合がある．TT-OOLS 方式では，k 人までの結託においてユーザを追跡可能にしようとするとセッション鍵を配布するのに必要なトラフィックが $O(k^4)$ にもなるので，MP3 のように比較的低ビットレートのコンテンツの配布にあたっては，それほど大きな k は設定できない．128 kbps の MP3 ファイルの場合，5 秒ごとにセッション鍵を再配布し，そのデータ転送量を全体の 10% 以内にしようとすると，k は 5 以下でなければならない．しかし，たとえば映像の放送のようにコンテンツが高ビットレートになると，相対的にセッション鍵の配布に必要なデータ転送コストが下がるため，k をより大きく設定できる．

5.1.3 実運用での評価

今回実装した MusicCast/AS サーバは，現在和歌山大学内 LAN で稼動状態である．クライアントは，学内 LAN に接続可能なコンピュータからクライアントプログラムを起動することでコンテンツを受信することができる．今回の動作実験では，LAN 内で 3 つのサブネットにわたってマルチキャストパケットを流通させながら，それぞれ 10 台程度での受信動作実験を行った．この動作実験において，サーバから配信されるコンテンツは，バッファリング処理による若干のずれはあるものの，マルチキャストで全クライアントが同時に同一のものを受信することが確認できた．また，放送時間中のクライアントの参加・脱退にもスムーズに対応できることが確認できた．マルチキャストでデータを配信すると，サーバからは単一のデータストリームだけが送信されることから，クライアント数によるサーバの負荷およびネットワーク上のトラフィックの面においてスケーラビリティが確保されているといえる．

一方，本実装は UDP を使用しているため，ネットワーク上で何らかの原因でパケットが欠落してしまうという事態は避けることができない．現状の MusicCast/AS では，このパケット欠落に対してはパリティパケットを使用して対処している．本実験の環境

☆ 今回は X Multimedia System（http://www.xmms.org/）を用いた．

はLAN内なのでパケットの欠落はまれにしか発生しない．そこでMusicCast/ASを長時間連続で使用すると，クライアントプログラムでパケットの欠落が確認できた．その場合，パケット欠落の発生する回数の大部分が単独の1パケットもしくは連続する2パケットの欠落であった．また，パケットの欠落が起こった回数の約半数でパリティパケットに補完されていることも確認できた．連続するパケットの欠落が複数回見られたことから，現在の実装のパリティによる誤り訂正はバースト的誤りに対処できないため，他の手法を採用するべきであることが分かった．

6. まとめ

マルチキャストを用いてスケーラブルに有料インターネット放送を行うシステムの機構について考察し，コンテンツの暗号化および配送にかかわる各種既存の方式を比較し，ネットワークに対するトラフィックの面と暗号化としてのセキュリティ耐性との両面から評価した結果，マルチキャスト通信におけるユーザ認証に応用した場合，比較的大人数のユーザに対する鍵配布にもスケーラブルに対応でき，運用上問題にならない程度のセキュリティを確保できることを示した．その結果をふまえ，このような有料インターネット放送が実現可能であることを示すプロトタイプとして，セッション鍵配送方式にTT-OOLS方式を採用した音楽放送システムMusicCast/ASを実装した．

今後の課題としては以下のようなものがあげられる．

- 今回はキャンパスLANにおいて実験を行ったが，今後広域での聴取実験を行い，実際のインターネット放送に近い環境での運用における知見を得たい．
- 現在は蓄積されたMP3ファイルを送信するのみのシステムになっているが，ライブ中継などにも利用できるよう実時間でMP3ファイルを生成するシステムにすることを計画している．また，動画像にも対応することを計画している．
- 現時点では配送プロトコルは独自であるが，広域分散環境で利用する際には，トラフィック監視などの管理を容易にするため現在広く使われているマルチメディアストリーミング向け配送プロトコルRTP[29]上で実装することが望ましい．またこの際，FECによるエラー訂正機能を強化するべきである．
- TT-OOLS法を使用した利点として，ユーザがすべて異なる認証用鍵でセッション鍵の復号化を行っている点がある．よって，セッション鍵の復号化からコンテンツの再生までを行うクライアントプログラムを何らかの形でブラックボックス化できれば，再生中のコンテンツに受信者のユーザID情報を電子透かしとして挿入することができ，受信者がコンテンツを不正に流用することを抑止できる．そのような研究を今後行いたい．

参考文献

1) RealNetworks Inc. HomePage. http://www.realnetworks.com/
2) Microsoft WindowsMedia HomePage. http://www.microsoft.com/windowsmedia/
3) Apple Computer Quicktime HomePage. http://www.apple.com/quicktime/
4) Deering, S.: Host Extensions for IP Multicasting, RFC1112 (1989).
5) Holbrook, H. and Cain, B.: Source-Specific Multicast for IP, Internet Draft (2001).
6) Deering, S. and Hinden, R.: Internet Protocol, Version 6 (IPv6) Specification, RFC2460 (1998).
7) Hinden, R. and Deering, S.: IP Version 6 Addressing Architecture, RFC2373 (1998).
8) Fenner, W.: Internet Group Management Protocol, Version 2, RFC2236 (1997).
9) Waitzman, D., Partridge, C. and Deering, S.: Distance Vector Multicast Routing Protocol, RFC1075 (1988).
10) Pusateri, T.: Distance Vector Multicast Routing Protocol, Internet Draft (2000).
11) Moy, J.: Multicast Extensions to OSPF, RFC1584 (1994).
12) Adams, A., Nicholas, J. and Siadak, W.: Protocol Independent Multicast–Dense Mode (PIM-DM): Protocol Specification (Revised), Internet Draft (2002).
13) Ballardie, A.: Core Based Trees (CBT version 2) Multicast Routing: Protocol Specification, RFC2189 (1997).
14) Estrin, D., Farinacci, D., Helmy, A., Thaler, D., Deering, S., Handley, M., Jacobson, V., Liu, C., Sharma, P. and Wei, L.: Protocol Independent Multicast–Sparse Mode (PIM-SM): Protocol Specification, RFC2362 (1998).
15) Deering, S., Fenner, W. and Haberman, B.: Multicast Listener Discovery (MLD) for IPv6, RFC2710 (1999).
16) The MPEG HomePage. http://mpeg.telecomitalialab.com/
17) ISO/IEC International Standard 11172-3: Coding of moving pictures and associated audio for digital storage media up to about 1,5 Mbits/s – Part 3: Audio (1993).

18) ISO/IEC International Standard 13818-3: Generic coding of moving pictures and associated audio information – Part 3: Audio (1998).
19) 高橋政雄：MP3 のデータ形式とその解析，Interface 2000 年 8 月号，pp.91–18, CQ 出版 (2000).
20) IETF The Group Security (GSEC) Research Group HomePage. http://securemulticast.org/gsec-index.htm
21) Andou, D., Sato, T., Hayashi, T., Tanabe, A., Izutsu, K., Goto, Y., Nishida, Y. and Inoue, W.: IGMP for user Authentication Protocol (IGAP), Internet Draft (2002).
22) Handley, M., Floyd, S., Whetten, B., Kermode, R., Vicisano, L. and Luby, M.: The Reliable Multicast Design Space for Bulk Data Transfer, RFC2887 (2000).
23) Harney, H. and Muckenhirn, C.: Group Key Management Protocol (GKMP) Specification, RFC2093 (1997).
24) Harney, H. and Muckenhirn, C.: Group Key Management Protocol (GKMP) Architecture, RFC2094 (1997).
25) Ballardie, A.: Scalable Multicast Key Distribution, RFC1949 (1996).
26) Wallner, D., Harder, E. and Agee, R.: Key Management for Multicast: Issues and Architectures, RFC2627 (1999).
27) Fiat, A. and Naor, M.: Broadcast Encryption, *Proc. Advances in Cryptology — Crypt'93*, pp.480–491 (1994).
28) Chor, B., Fiat, A., Naor, M. and Pinkas, B.: Tracing Traitors, *IEEE Trans. Information Theory*, Vol.46, No.3, pp.893–910 (2000).
29) Schulzrinne, H., Casner, S., Frederick, R. and Jacobson, V.: RTP: A Transport Protocol for Real-Time Applications, RFC1889 (1996).

(平成 14 年 7 月 11 日受付)
(平成 14 年 12 月 3 日採録)

上原哲太郎（正会員）

1990 年京都大学工学部情報工学科卒業．1992 年同大学大学院修士課程修了．1995 年同大学院博士後期課程研究指導認定退学．同年同大学院工学研究科助手．1996 年和歌山大学情報処理センター講師．1997 年同大学システム情報学センター講師．2000 年同大学システム工学部情報通信システム学科講師，現在に至る．自動並列化コンパイラ，分散並列処理，システム運用技術，インターネットセキュリティ等の研究に従事．京都大学博士（工学）．日本ソフトウェア科学会，CIEC 各会員．

川北 良一

2000 年和歌山大学システム工学部情報通信システム学科卒業．2002 年同大学大学院修士課程修了．現在，ネットワンシステムズ株式会社に勤務，ネットワークインテグレーション業務に従事．修士（システム工学）．

辻 義一

2002 年和歌山大学システム工学部情報通信システム学科卒業．現在，TIS 株式会社に勤務，プラットフォームやネットワークの設計･構築に従事．

佐藤 敬

1991 年東京工業大学工学部電気電子工学科卒業．1993 年同大学大学院理工学研究科修士課程修了．1994 年同大学院理工学研究科博士後期課程中退．同年同大学工学部助手．1999 年北九州大学国際環境工学部設置準備室講師．2001 年北九州市立大学国際環境工学部助教授，現在に至る．情報セキュリティ，コンピュータネットワークの研究に従事．博士（工学）．電子情報通信学会，IEEE，IACR 各会員．

山岡　克式（正会員）

1991年東京工業大学工学部電気電子工学科卒業．1993年同大学大学院理工学研究科修士課程修了．1994年同大学院理工学研究科博士後期課程退学．同年東京工業大学工学部助手．2000年文部省メディア教育開発センター研究開発部助教授．2001年東京工業大学学術国際情報センター助教授，現在に至る．電話網，コンピュータネットワークの両面にわたり，ネットワークのメディアQoS制御，ネットワーク動的分散制御，情報検索に関する通信制御，メディア同期制御，コンテンツ流通制御等の研究に従事．博士（工学）．電子情報通信学会会員．

泉　　裕

1993年和歌山大学教育学部情報科学学科卒業．1995年奈良先端科学技術大学院大学博士前期課程修了．1998年同大学院大学博士後期課程単位取得満期退学．同年和歌山大学システム情報学センター助手，現在に至る．ネットワークアーキテクチャ，ネットワーク管理，インターネットセキュリティ等の研究に従事．修士（工学）．ISOC会員．

齋藤　彰一（正会員）

1993年立命館大学理工学部情報工学科卒業．1995年同大学大学院博士前期課程修了．1998年同大学院博士後期課程単位取得満期退学．同年和歌山大学システム工学部情報通信システム学科助手，現在に至る．オペレーティングシステム，分散並列処理，インターネット等の研究に従事．博士（工学）．日本ソフトウェア科学会，ACM，IEEE-CS各会員．

國枝　義敏（正会員）

1980年京都大学工学部情報工学科卒業．1982年同大学大学院修士課程修了．同年京都大学工学部情報工学科助手．1991年同助教授．1996年和歌山大学システム工学部情報通信システム学科教授，現在に至る．工学博士．主として，計算機ソフトウェア，システムプログラム，言語処理系，超高速計算等の分野に関する研究に従事．電子情報通信学会，ACM，IEEE-CS各会員．

結城　皖曠

1964年大阪府立大学工学部電気工学科卒業．同年日本電信電話公社（現NTT）入社，研究所配属．以来，PCM端局装置，ファクシミリ伝送方式，ファクシミリ通信網の研究実用化に従事．NTT-AT株式会社取締役を経て現在メディア教育開発センター研究開発部教授．教育ネットワークシステムの研究に従事．工学博士．電子情報通信学会，映像情報メディア学会，画像電子学会各会員．1980年度電子情報通信学会論文賞，1983年度同学会業績賞受賞．