放送型コンテンツ配信における匿名視聴方式の提案

櫻井友二[†] 齊藤泰一[††]

従来の動画コンテンツ配信において，コンテンツプロバイダがユーザに対して課金を行う場合，いずれのユーザがどのようなチャネル/番組を視聴したのかというプライバシ情報をコンテンツプロバイダは知りえてしまう．本論文では，チャネル単位課金/番組単位課金であるような放送型番組配信におけるユーザのプライバシを保護した匿名視聴方式を提案する．

A Charging Method without Disclosing Channel Information

Yuji Sakurai[†] and Taiichi Saito[††]

This paper proposes a charging method without disclosing channel information. In contents delivery systems, content provider charges a user with a fee based on his/her choice of channels/programs. Then, the provider could know the user's privacy information such as channel that the user watched. In this paper, we propose a contents delivery system that protects the user's privacy information.

1. はじめに

近年，インターネット，衛星放送，地上波放送，携帯電話などの普及により，動画コンテンツ配信サービスの利用が増加している．コンテンツの配信形態は，コンテンツプロバイダがコンテンツを一方的にユーザへ向け配信し続けそのコンテンツをユーザ側が受信して視聴する放送型と，ユーザが視聴したいコンテンツをコンテンツプロバイダにリクエストして希望したコンテンツのみを取得して視聴するオンデマンド型に分けることができる．

動画コンテンツ視聴の料金体系としては，見ることができる番組があらかじめ決まっているチャネルを購入して視聴するチャネル単位課金，見たい番組のみを購入して視聴する番組単位課金などがある．番組単位課金の例としては，Pay-Per-View（PPV）がある．PPV とは，ユーザが視聴した分だけ料金を払うテレビ番組の課金システムである．

ここで，従来の動画コンテンツ配信で問題となるのがユーザのプライバシである．コンテンツプロバイダは，ユーザがいずれのチャネル/番組を視聴したのかという情報に基づいてユーザに対して利用料金の請求を行う．コンテンツプロバイダは，ユーザが視聴したチャネル/番組が分からなければ，ユーザに対して正しく利用料金を請求できない．すなわち，コンテンツプロバイダはユーザがいずれのチャネル/番組を視聴したのか知る必要があるのである．またそのことにより，コンテンツプロバイダはいずれのチャネル/番組が視聴されているかという視聴情報を同時に得ることができる．各チャネル/番組の視聴情報は，コンテンツプロバイダにとっては有益な情報であると考えられる．

しかし，ユーザは視聴したチャネル/番組をコンテンツプロバイダやその他の第三者に知られたくない場合もあると考えられる．また，個人情報に対する意識が高まっている昨今，ユーザがどのようなチャネル/番組を視聴しているのかといった情報も個人情報の 1 つとして慎重に扱わなければならない．したがって，ユーザの視聴履歴，視聴傾向はプライバシの観点からコンテンツプロバイダや第三者に対して秘密にしなければならないと考えられる．

ユーザがどのようなチャネル/番組を視聴しているのかといった情報をコンテンツプロバイダに対して秘匿した場合，コンテンツプロバイダはユーザに対して正当な利用料金を請求できなくなる．また，視聴情報を得ることができなくなる．よって，以下のことが求められている．

† 東京電機大学大学院工学研究科情報通信工学専攻
Graduate School of Engineering, Tokyo Denki University
†† 東京電機大学工学部情報通信工学科
Department of Information and Communication Engineering, Tokyo Denki University

- コンテンツプロバイダは，ユーザが視聴したチャネル/番組は分からないが，ユーザに対して正しく利用料金を請求できること
- 各チャネル/番組の視聴情報は，コンテンツプロバイダにとっては有益な情報であるため，ユーザの情報を秘匿したまま視聴情報をコンテンツプロバイダが取得できること

ここで守らなければならない情報は，「視聴時においてユーザがいずれのチャネル/番組を視聴したかという情報」と「コンテンツプロバイダの視聴情報取得時において視聴情報に対応したチャネル/番組をいずれのユーザが視聴したかという情報」である．

本論文では，上記 2 つの要求を満足し，さらにチャネル/番組ごとに異なる料金を設定できることを可能にする，放送型コンテンツ配信におけるチャネル単位課金/番組単位課金の，ユーザのプライバシを保護した匿名視聴方式の提案を行う．

1.1 提案方式を可能にする技術

以下では，提案方式を構成する主な既存技術である Oblivious Transfer，Secret Sharing scheme，Mixnet について簡単に説明する．

1.1.1 Oblivious Transfer

Oblivious Tansfer（OT）は最初に Rabin によって提案された[1]．後に，様々な種類が提案されている．例として，1-out-of-2 OT[2]，1-out-of-n OT[3]，k-out-of-n OT[4]，adaptive k-out-of-n OT[5] などがある．ここでは提案方式に用いる 1-out-of-n OT について説明する．

1-out-of-n OT は Brassard らによって提案され[3]，その後 Julien らにより効率的な方式が提案されている[6]．1-out-of-n OT における参加者は，送信者と受信者とする．送信者が n 個の情報 $m_0, m_1, \ldots, m_{n-1}$ を保持しており，そのうちいずれか 1 つのみを受信者は受け取ることができるが，受信者が $m_0, m_1, \ldots, m_{n-1}$ のうちのいずれを受け取ったかを送信者は知ることができない．一方受信者は，受け取らなかった情報についてはいっさい知ることはできない．

ここで，提案方式における 1-out-of-n OT のプロトコルを定義する．送信者と受信者が存在するとして，受信者は送信者が保持する情報 $m_1, m_2, \ldots, m_n$ の中から i 番目の情報 m_i を得たいとする．送信者に対する情報 m_i の問合せを q とし，q を計算する関数を $Query$ とする．受信者は，i と乱数 r_i（r_i は情報 m_i の問合せに必要な乱数）を関数 $Query$ の入力とし，問合せ $q = Query(i, r_i)$ を得る．受信者は，情報 m_i の問合せ q を送信者に対して送る．受信者からの問合せ q に対する応答 a を計算する関数を $Answer$ とする．受信者から送られてきた問合せ q と送信者が保持する情報 $m_1, m_2, \ldots, m_n$ を関数 $Answer$ の入力とし，$a = Answer(q, (m_1, m_2, \ldots, m_n))$ を得る．送信者は応答 a を受信者に送る．応答 a から情報 m_i のみを得る関数を $Retrieve$ とする．受信者は送信者からの応答 a と r_i を $Retrieve$ の入力とし，$m_i = Retrieve(a, r_i)$ を得る．関数 $Retrieve$ は応答 a の中から情報 m_i のみを取り出す．

1-out-of-n OT は以下の性質を満たさなければならない．

- 受信者からの問合せ q から i の情報は漏れない
- 送信者からの応答 a から m_i 以外の情報は漏れない

提案方式では，1-out-of-n OT を用いることで，いずれのチャネルを視聴したかをだれにも知られることなくユーザは各チャネルを視聴することができる．

1.1.2 Secret Sharing scheme

Secret Sharing scheme（SS：秘密分散法）は，Shamir[7] と Blakley[8] により独立に提案された．その後さかんに研究され，分散情報が確かにその秘密から生成されたものであるかが検証可能な Verifiable Secret Sharing scheme（VSS：検証可能秘密分散法）などの応用が提案されている．VSS においては，Pedersen[9] と Feldman[10] が暗号学的仮定を導入することで，非対話的（non-interactive）でかつ効率的な構成法を提案している．

SS は，秘密をあらかじめ複数の分散情報に分割して保管し，必要なときにその分散情報を持ち寄って秘密を復元する暗号技術である．特に，k-out-of-n SS では，秘密情報 s を n 個の分散情報に符号化し，そのうち任意の k 個の分散情報を集めれば元の秘密情報 s が完全に復元できるが，いずれの $k-1$ 個の分散情報からでは s を復元することはできない．

これより，提案方式に用いる VSS を Feldman の方式に基づいて定義する．VSS は，3 つの関数，情報分散関数 $Share$ と情報復号関数 $Merge$，分散情報検証関数 $Verify$ から構成される．秘密情報を s とすると，s より n 個の分散情報を生成，出力し，さらにその中のある 1 つの分散情報が確かに s の分散情報であることを検証するために必要な値（検証情報）を出力する関数を情報分散関数 $Share$ とする．k を秘密情報 s を復元するのに必要な分散情報のしきい値，n を分散情報の総数，$(s_1, s_2, \ldots, s_n)$ を分割された n 個の分散情報，v_s を s についての検証情報とする．秘密情報 s，しきい値 k，分散情報の総数 n，これら 3 つを関数

Share の入力とすることで，s の復元に必要なしきい値を k に設定し，s について n 個の分散情報とその検証情報 $(s_1, s_2, \ldots, s_n, v_s) = Share(s, k, n)$ を得る．s_i $(1 \leq i \leq n)$ が確かに s の分散情報であることを検証する関数を分散情報検証関数 *Verify* とする．関数 *Verify* は，s_i と v_s を関数 *Verify* の入力とし，s_i が s の分散情報であるときには $accept = Verify(s_i, i, v_s)$ を，そうでない場合には $reject = Verify(s_i, i, v_s)$ を出力する．上記の n 個ある分散情報の中から選択した k 個の分散情報 $(s_{t_1}, s_{t_2}, \ldots, s_{t_k})$ より，秘密情報 s を復元する関数を情報復号関数 *Merge* とする．k 個の分散情報 $(s_{t_1}, s_{t_2}, \ldots, s_{t_k})$，各分散情報のインデックス $\{t_1, t_2, \ldots, t_k\} \subset \{1, 2, \ldots, n\}$，しきい値 k，分散情報の総数 n，これら 4 つを関数 *Merge* の入力とし，$s = Merge((s_{t_1}, s_{t_2}, \ldots, s_{t_k}), (t_1, t_2, \ldots, t_k), k, n)$ を得る．

VSS は以下の性質を満たさなければならない．

- いずれの $k-1$ 個以下の分散情報から秘密情報 s を復元することはできない．

- k 個の分散情報から正しく s を復元できなかった場合，s の分配者の不正を検出することができる．

提案方式では，VSS を用いることで各チャネルに対して異なる価格を設定することが可能となる．

1.1.3 Mixnet

Mixnet [11] は，複数のサーバを介して匿名通信路を実現する技術である．Mixnet は，利用者に対して匿名性を提供することでプライバシを保護するが，その一方で利用者がその匿名性を悪用して不正を行うことを可能としてしまう．近年では，千田らによって利用者の匿名性の確保と，利用者が不正を行った場合にはその不正利用者を特定することができる方式が提案されている [12]．提案方式では，ユーザの匿名性の確保ができれば十分であるため，匿名通信のみが可能である Mixnet を利用することとする．

送信者 S と受信者 R が，Mixnet を用いて S がメッセージ msg を受信者 R に送ることを，$mix(R, msg)$ と書くとする．

1.2 記号の定義

提案方式を構成するために共通鍵暗号アルゴリズム，署名アルゴリズムを以下のように定義する．提案方式ではこれらのアルゴリズムを用いる．

- 共通鍵暗号アルゴリズム

平文を m，暗号化アルゴリズムを E，復号アルゴリズムを D，暗号化鍵を K，復号鍵を K' とする．暗号化において，平文 m を暗号化し暗号文 c を得ることを，$c = E(K, m)$ と書くとする．また復号において，暗号文 c を復号して平文 m を得ることを，$m = D(K', c)$ と書くとする．

- 署名アルゴリズム

平文を m，鍵生成アルゴリズムを G，署名生成アルゴリズムを S，検証アルゴリズムを V，署名鍵と検証鍵のペアをそれぞれ K_{sig}，K_{ver} とする．鍵生成において，鍵生成アルゴリズム G と乱数 r から署名鍵 K_{sig} と検証鍵 K_{ver} を得ることを，$(K_{sig}, K_{ver}) = G(r)$ と書くとする．署名生成において，平文 m に対する署名 σ を得ることを，$\sigma = S(m, K_{sig})$ と書くとする．また検証において，σ が平文 m に対して署名鍵 K_{sig} を用いて生成された署名であるとき，$accept = V(m, \sigma, K_{ver})$，そうでないときは，$reject = V(m, \sigma, K_{ver})$ がそれぞれ出力されるとする．

1.3 関 連 研 究

1.3.1 放送型暗号

正規ユーザのみを視聴可能とする有料放送サービスの実現方法として，放送型暗号（broadcast encryption）が提案されている [13]．放送を行うセンタは，セッション鍵を用いて暗号化されたデータをユーザに送信し，受信したユーザはあらかじめ個人ごとに固有に与えられた鍵（個人鍵）を用いて復号を行う．近年では，光成らによって楕円曲線上のペアリングを用いた不正者追跡機能，鍵漏洩の自己抑止力，非対称不正者追跡機能，および加入者排除機能を持つ方式が提案されている [14]．放送型暗号では，ユーザはセンタから放送されているセッション鍵のヘッダ情報と自身が持つ個人鍵からセッション鍵を復号して取得する．

提案方式は，放送型暗号におけるセッション鍵そのものを秘密分散法を用いて複数の分散情報に分割し，そのセッション鍵を復元するのに必要な数の分散情報を取得し，それらからセッション鍵を復元して取得するという点で異なる．また，放送型暗号ではユーザに対して課金する仕組みについては考慮されていない．

1.3.2 Priced Oblivious Transfer

Priced Oblivious Transfer（Priced OT）は，商品購入者がいずれの商品を購入したかを秘匿しながら商品を販売するプロトコルで，Aiello らによって提案された [15]．Priced OT では，購入者はあらかじめ上限価格のチケットを購入しておき，その上限に達するまでチケットの購入者と購入商品の組合せを知られずにコンテンツを購入することができる．

しかし，各コンテンツの価格によっては，購入商品の組合せを知られないようにするために上限価格をある程度高く設定しなければならないと考えられる．高額な金額を事前に販売者に対して支払わなければなら

ないとき，そのことは購入者にとって負担となる可能性がある．また，購入者が上限価格を使いきらない場合には，販売者は残りの金額を購入者に返金する処理が必要となるという運用上の問題がある．

1.3.3 利用履歴を秘匿できるコンテンツ配信・課金方式

飛田らは，Ateniese らにより提案されたグループ署名を利用し，ユーザのコンテンツの利用履歴や利用傾向を秘匿できるコンテンツ配信・課金方式（CDCS）を提案している[16]．CDCS では，ユーザの計算・通信コストが利用可能なコンテンツの総数に依存せず利用したコンテンツ数に依存する．CDCS を放送型として構成することで，提案方式と同様のプライバシ保護が実現可能であると考えられる．

提案方式は，ユーザの計算・通信コストは配信されるコンテンツの総数と視聴するコンテンツの価格に依存する点，ユーザへの課金方法が違う点，信頼できる第三者機関を利用しないという点で CDCS とは異なる．また CDCS は，ユーザの一定期間における利用金額の合計をたかだか 10^5 程度と仮定しているのに対し，提案方式ではユーザの利用金額の合計は無制限である．

2. 匿名視聴方式の提案

これより匿名視聴方式の提案を行う．従来の放送型コンテンツ配信に求められている要求は，「コンテンツプロバイダは，ユーザが視聴したチャネル/番組は分からないが，ユーザに対して正しく利用料金を請求できること」，「各チャネル/番組の視聴情報は，コンテンツプロバイダにとっては有益な情報であるため，ユーザの情報を秘匿したまま視聴情報をコンテンツプロバイダが取得できること」，これら 2 つであった．提案方式は，これら 2 つの要求を満足し，さらにチャネル/番組ごとに異なる料金を設定できることを可能にする．

ここで，チャネルおよび番組について定義する．はじめに，チャネルについて定義する．1 つのチャネルは複数の番組より構成され，各チャネルの番組構成はコンテンツプロバイダによってあらかじめ決められているとする．チャネルの契約期間については，1 日ごと，1 カ月ごとなどある程度時間がまとまった一定期間とする．提案方式では，チャネル数は 50 程度を想定する．

次に番組について定義する．1 つの番組は，数十分から数時間程度の時間的に連続したコンテンツとする．番組の契約期間については，契約した番組が開始された時点からその番組が終了した時点までとする．提案方式では，番組数は 5,000 程度を想定する．

上記の定義より，1 つの番組はきわめて契約期間が短いチャネルであると見なすことができる．したがって，以下ではチャネル単位課金についてのみ考えることとする．

提案方式は，いずれのチャネルを視聴したのかコンテンツプロバイダに知られることなくユーザはチャネルを視聴することができ，各チャネルごとに異なる価格設定が可能で，コンテンツプロバイダが視聴情報を取得可能な方式である．提案方式では，複数のデータベースを利用し，Oblivious Transfer，VSS，Mixnet を用いる．Oblivious Transfer を用いることで，いずれのチャネルを視聴したのかコンテンツプロバイダに知られることなくユーザはチャネルを視聴することができる．また VSS を用いることで，コンテンツプロバイダが提供する各チャネルに対して異なる価格を設定することが可能である．さらに Mixnet を用いることで，コンテンツプロバイダはいずれのチャネルが視聴されているのかといった視聴情報を得ることができる．提案方式では，ユーザにおける計算・通信コストは配信されるチャネルの総数と視聴するチャネルの価格に依存する．

2.1 提 案 方 式

提案方式の参加者は，ユーザ U，分散情報データベース $DB_1, DB_2, \ldots, DB_n$，放送局 BS とする．U と $DB_1, DB_2, \ldots, DB_n$ は通信を行うことが可能であるとし，Mixnet は匿名通信路であるとする．

提案方式は，セットアップフェーズ，復号鍵復元フェーズ，視聴フェーズ，請求フェーズから構成される．また提案方式において，各チャネルと視聴情報を配信する仕組み，提案方式での取り決め，ユーザの加入・登録処理は以下のとおりである．

- 各チャネルと視聴情報を配信する仕組み
  - 各チャネルは共通鍵暗号を用いて異なる暗号化鍵で暗号化されている．
  - チャネル cha_j の暗号化は，cha_j の暗号化鍵 K_j と cha_j を共通鍵暗号アルゴリズムの暗号化アルゴリズム E の入力とし，暗号化された cha_j を $c_j = E(K_j, cha_j)$ として得る．
  - 暗号化された各チャネルはすべてのユーザへブロードキャストされている．
  - 放送局 BS は，署名アルゴリズムの鍵生成アルゴリズム G と乱数 r_{BS} を用いて，BS 自身の署名鍵と検証鍵のペア $(K_{sig}^{BS}, K_{ver}^{BS}) = G(r_{BS})$ を生成する．

- チャネル cha_j が視聴されたという情報を，視聴情報とし $view_j$ と書くとする．
- BS は，署名生成アルゴリズム S と自身の署名鍵 K_{sig}^{BS} を用いて，$view_j$ に対する署名 $\sigma_{view_j}^{BS} = S(view_j, K_{sig}^{BS})$ を得る．
- BS の署名付き視聴情報 $\sigma_{view_j}^{BS}$ の暗号化は，cha_j の暗号化鍵である K_j と $\sigma_{view_j}^{BS}$ を共通鍵暗号アルゴリズムの暗号化アルゴリズム E の入力とし，暗号化された $\sigma_{view_j}^{BS}$ を $c\sigma_{view_j}^{BS} = E(K_j, \sigma_{view_j}^{BS})$ として得る．
- 暗号化された各チャネルの BS の署名付き視聴情報は，暗号化された各チャネルと同様にすべてのユーザへブロードキャストされている．

- 提案方式での取り決め
  - 放送局 BS がユーザ U に提供するチャネルの総数を m とする．
  - BS は n 個の分散情報データベース DB_1, $DB_2, \ldots, DB_n$ を用意する．
  - 各データベース DB_i（$1 \leq i \leq n$）は，署名アルゴリズムの鍵生成アルゴリズム G と乱数 r_{DB_i} を用いて，自身の署名鍵と検証鍵のペア $(K_{sig}^{DB_i}, K_{ver}^{DB_i}) = G(r_{DB_i})$ を生成する．
  - DB_i は，検証鍵 $K_{ver}^{DB_i}$ を BS が用意したデータベース検証鍵リストに登録する．
  - 分散情報データベースから 1 回の分散情報の取得にかかる料金はすべて同一であるとする．
  - 必要な数の分散情報を集めるとチャネルの復号鍵を復元できる．
  - BS は一定の期間を定め（例として 1 カ月など），その一定期間に U が取得した分散情報の総数である分散情報総取得回数と 1 回の分散情報の取得にかかる料金とを掛け合わせたものを利用料金として U に対して請求する．
- ユーザの加入・登録処理
  - ユーザ U は，署名アルゴリズムの鍵生成アルゴリズム G と乱数 r_u を用いて，ユーザ自身の署名鍵と検証鍵のペア $(K_{sig}^u, K_{ver}^u) = G(r_u)$ を生成し，検証鍵 K_{ver}^u を各分散情報データベース DB_i に登録する．
  - DB_i は，U の検証鍵登録時に U の分散情報取得回数（number of key acquisition）$NKA_{DB_i}^u$ を生成し登録・管理する（分散情報取得回数 $NKA_{DB_i}^u$ の初期値は 0 とする）．
  - U は，放送局 BS からデータベース検証鍵リストを取得する．

【セットアップフェーズ】

放送局 BS は，1 回の分散情報の取得にかかる料金 y と，チャネル cha_j（$1 \leq j \leq m$）の価格が $x_j = k_j y$（$2 \leq k_j \leq n$）となるようにしきい値 k_j を設定する．BS は，x_j と y をユーザ U に対して公開する．BS は，K_j について関数 $Share$ を用い n 個の分散情報とその検証情報を計算する．BS は，cha_j の復号鍵 K_j，しきい値 k_j，分散情報の総数 n，これら 3 つを関数 $Share$ に入力し，n 個の分散情報とその検証情報 $(K_j^1, K_j^2, \ldots, K_j^n, v_{K_j}) = Share(K_j, k_j, n)$ を得る．

次に，BS は検証情報 v_{K_j} を U に対して公開し，分散情報 $(K_j^1, K_j^2, \ldots, K_j^n)$ を用意した n 個の分散情報データベース $DB_1, DB_2, \ldots, DB_n$ にそれぞれ秘密裏に格納する．各分散情報は，分散情報データベースにそれぞれ以下のように格納される．

—分散情報 K_j^1 は分散情報データベース DB_1 へ格納．
—分散情報 K_j^2 は分散情報データベース DB_2 へ格納．
⋮
—分散情報 K_j^n は分散情報データベース DB_n へ格納．

BS は，この操作を提供する m のチャネルすべてについて行う．ここで，DB_i に格納される分散情報は，

—チャネル cha_1 の復号鍵 K_1 の分散情報 K_1^i
—チャネル cha_2 の復号鍵 K_2 の分散情報 K_2^i
⋮
—チャネル cha_m の復号鍵 K_m の分散情報 K_m^i

となる．

以下では，U が暗号化されブロードキャストされている価格が $x_j = k_j y$ のチャネル cha_j を復号して視聴したい場合を考える．この場合，U は k_j 個のデータベースにアクセスすることになる．

【復号鍵復元フェーズ】

1. ユーザ U は，n 個の分散情報データベースから k_j 個の分散情報データベース $(DB_{l1}, DB_{l2}, \ldots, DB_{lk_j})$ を選択する（ただし $\{DB_{l1}, DB_{l2}, \ldots, DB_{lk_j}\} \subset \{DB_1, DB_2, \ldots, DB_n\}$ とする）．

ここでは，分散情報データベース DB_{lk_j}（$DB_{lk_j} \in \{DB_1, DB_2, \ldots, DB_n\}$）と U との間のプロトコルを示す．U は，cha_j の分散情報を得るために，j と乱数 r_j を関数 $Query$ の入力とし，問合せ $q_j^u = Query(j, r_j)$ を得る．また U は，署名アルゴリズムの署名生成アルゴリズム S と自身の署名鍵 K_{sig}^u を用いて，q_j^u に対する署名 $\sigma_{q_j}^u = S(q_j^u, K_{sig}^u)$ を生成する．そして，q_j^u と $\sigma_{q_j}^u$ を DB_{lk_j} に送る．

2. DB_{lk_j} は，U から送られてきた問合せ q_j^u，署名 $\sigma_{q_j}^u$，自身が管理している U の検証鍵 K_{ver}^u を

署名アルゴリズムの検証アルゴリズム V の入力としその結果を得ることで，U が正規のユーザであるか検証する．$reject = V(q_j^u, \sigma_{q_j}^u, K_{ver}^u)$ の場合は処理を終了する．$accept = V(q_j^u, \sigma_{q_j}^u, K_{ver}^u)$ の場合は，q_j^u と DB_{lk_j} が保持する各チャネルの復号鍵の分散情報を関数 $Answer$ の入力として，応答 $a_j^{DB_{lk_j}} = Answer(q_j^u, (K_1^{lk_j}, K_2^{lk_j}, \ldots, K_m^{lk_j}))$ を計算する．また DB_{lk_j} は，署名アルゴリズムの署名生成アルゴリズム S と自身の署名鍵 $K_{sig}^{DB_{lk_j}}$ を用いて，$a_j^{DB_{lk_j}}$ に対する署名 $\sigma_{a_j}^{DB_{lk_j}} = S(a_j^{DB_{lk_j}}, K_{sig}^{DB_{lk_j}})$ と $\sigma_{q_j}^u$ に対する署名 $\sigma_{\sigma_{q_j}^u}^{DB_{lk_j}} = S(\sigma_{q_j}^u, K_{sig}^{DB_{lk_j}})$ をそれぞれ計算する．DB_{lk_j} は，$a_j^{DB_{lk_j}}$，$\sigma_{a_j}^{DB_{lk_j}}$，$\sigma_{\sigma_{q_j^u}}^{DB_{lk_j}}$ を U へ送る．

ここで DB_{lk_j} は，U がいずれのチャネルの復号鍵の分散情報を 1 つ取得したため，U の分散情報取得回数 $NKA_{DB_{lk_j}}^u$ を 1 増やし，$\sigma_{q_j}^u$ を保存しておく．

3. U は，自身が生成した q_j^u に対する署名 $\sigma_{q_j}^u$，DB_{lk_j} から送られてきた応答 $a_j^{DB_{lk_j}}$ と 2 つの署名 $\sigma_{a_j}^{DB_{lk_j}}$，$\sigma_{\sigma_{q_j^u}}^{DB_{lk_j}}$，自身が保持しているデータベース検証鍵リストにある DB_{lk_j} の検証鍵 $K_{sig}^{DB_{lk_j}}$ を署名アルゴリズムの検証アルゴリズム V の入力とし，それぞれが $accept = V(a_j^{DB_{lk_j}}, \sigma_{a_j}^{DB_{lk_j}}, K_{ver}^{DB_{lk_j}})$，$accept = V(\sigma_{q_j}^u, \sigma_{\sigma_{q_j^u}}^{DB_{lk_j}}, K_{ver}^{DB_{lk_j}})$ を出力した場合，$DB_{lk_j} \in \{DB_{lk_1}, DB_{lk_2}, \ldots, DB_{lk_j}\}$ であることが確認できる．

次に U は，$a_j^{DB_{lk_j}}$ と乱数 r_j を関数 $Retrieve$ の入力とし，cha_j の復号鍵 K_j の分散情報 $K_j^{lk_j} = Retrieve(a_j^{DB_{lk_j}}, r_j)$ を得る．そして，U は $K_j^{lk_j}$ が確かに cha_j の復号鍵 K_j の分散情報であることを検証する．U は，$K_j^{lk_j}$ と放送局 BS によって公開されている検証情報 v_{K_j} を関数 $Verify$ の入力とし，$accept = Verify(K_j^{lk_j}, j, v_{K_j})$ が出力された場合，$K_j^{lk_j}$ が確かに K_j の分散情報であることを確認できる．

このようにして，1 つの分散情報データベースから 1 つの分散情報のみを得る．上記の操作を，選択した k_j 個の分散情報データベースすべてについて行い，K_j を復元するのに必要な k_j 個の分散情報 $K_j^{l1}, K_j^{l2}, \ldots, K_j^{lk_j}$ を得る．

U は，取得した k_j 個の分散情報，各分散情報のインデックス $\{l1, l2, \ldots, lk_j\} \subset \{1, 2, \ldots, n\}$，しきい値 k_j，分散情報の総数 n，これら 4 つを関数 $Merge$ の入力とし，$K_j = Merge((K_j^{l1}, K_j^{l2}, \ldots, K_j^{lk_j}), (l1, l2, \ldots, lk_j), k_j, n)$ を得る．セットアップフェーズと復号鍵復元フェーズを**図 1** に示す．

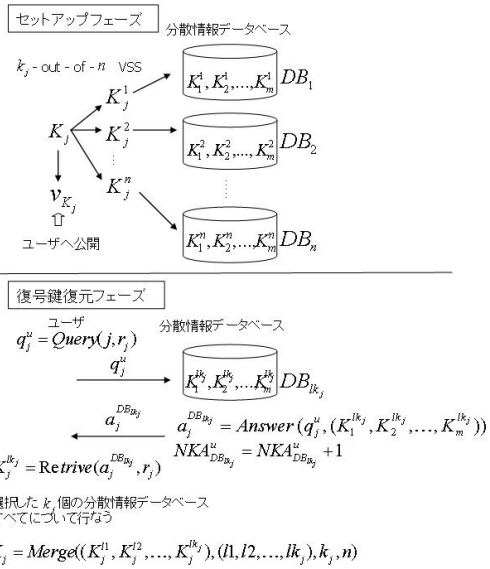

図 1 セットアップフェーズと復号鍵復元フェーズ
Fig. 1 Setup phase and reconstructing a decryption key phase.

【視聴フェーズ】

c_j は，放送局 BS によって共通鍵暗号アルゴリズムの暗号化アルゴリズムで暗号化されたチャネル cha_j である．また $c\sigma_{view_j}^{BS}$ は，BS によって共通鍵暗号アルゴリズムの暗号化アルゴリズムで暗号化されたチャネル cha_j の BS の署名付き視聴情報である．ユーザ U は，復号鍵復元フェーズで得た cha_j の復号鍵 K_j と BS からブロードキャストされている c_j を共通鍵暗号アルゴリズムの復号アルゴリズム D の入力とし，$cha_j = D(K_j, c_j)$ を復号し視聴する．さらに U は，K_j と BS からブロードキャストされている $c\sigma_{view_j}^{BS}$ を共通鍵暗号アルゴリズムの復号アルゴリズム D の入力とし，$\sigma_{view_j}^{BS} = D(K_j, c\sigma_{view_j}^{BS})$ を復号し取得する．そして U は，$\sigma_{view_j}^{BS}$ を Mixnet を利用して BS へ送信する．$mix(BS, \sigma_{view_j}^{BS})$ より，BS は $\sigma_{view_j}^{BS}$ を取得することができる．

【請求フェーズ】

各データベース DB_i（$1 \leq i \leq n$）が保持するユーザ U の分散情報取得回数 $NKA_{DB_i}^u$ は，定められた一定の期間内に U が分散情報データベース DB_i より取得した復号鍵の分散情報の個数である．放送局 BS は，各 DB_i からそれらが保持する $NKA_{DB_i}^u$ と U の問合せに対する署名☆を取得し，$NKA_{DB_i}^u$ と

☆ 問合せに対する署名は，$\sigma_{q_{j1}}^u, \sigma_{q_{j2}}^u, \ldots, \sigma_{q_{jNKA_{DB_i}^u}}^u$（$\{j1, j2, \ldots, jNKA_{DB_i}^u\} \subset \{1, 2, \ldots, m\}$）とする．

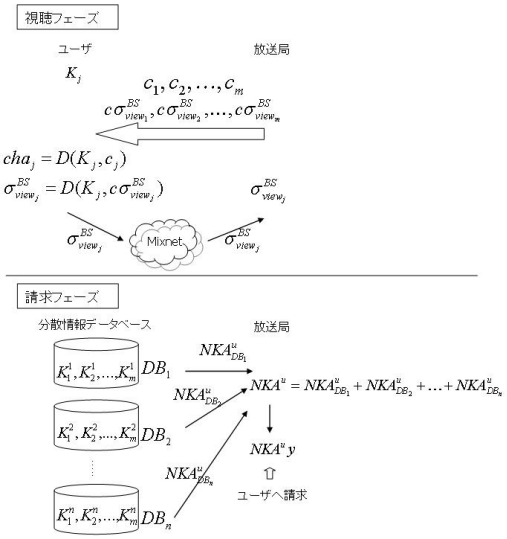

図 2 視聴フェーズと請求フェーズ
Fig. 2 Viewing phase and charging phase.

U の問合せに対する署名の総数が等しくなることを確認する．次に，BS は U の分散情報総取得回数 $NKA^u = NKA^u_{DB_1} + NKA^u_{DB_2} + \ldots + NKA^u_{DB_n}$ を計算する．そして，NKA^u と 1 回の分散情報の取得にかかる料金 y とを掛け合わせた金額 $NKA^u y$ を利用料金として U へ請求する．利用料金請求後，各 DB_i は自身が管理する $NKA^u_{DB_i}$ を初期値 0 に戻し，保持する U の問合せに対する署名をすべて破棄する．視聴フェーズと請求フェーズを**図 2** に示す．

3. 考　　察

3.1 提案方式の特徴

ここで，提案方式の特徴を整理する．

ユーザの匿名視聴が可能：ユーザ U は，各チャネル cha_j（$1 \leq j \leq m$）をだれにも知られることなく視聴することが可能である．さらに，放送局 BS は U がいずれのチャネルを視聴したかを知ることなしに，U に対して正当な利用料金を請求することができる．

各チャネルに異なる価格を設定可能：BS は，暗号化された各チャネル cha_j の復号鍵 K_j を復元するのに必要な分散情報のしきい値 k_j と 1 回の分散情報の取得にかかる料金 y を自由に設定できるため，各チャネルに対して異なる価格を決定することができる．ただし，cha_j の料金 $k_j y$ は使用するデータベース数が n である場合，$2y \leq k_j y \leq ny$ の範囲でなければならない．

視聴情報の取得が可能：BS は，U がいずれのチャネルを視聴したかを知ることなく，チャネルがいずれかのユーザに視聴されたという視聴情報のみを得ることができる．ただし，U が視聴情報を放送局に対して送信しなかった場合，BS は U に対して視聴情報の送信を強制することはできない．したがって，BS は視聴情報を U の協力を得ることで取得することができる．

2 つの課金方式：BS は，チャネル単位および番組単位で U に対して課金を行えるため，U のニーズに広く対応し，柔軟な課金を行うことができる．チャネル単位課金と番組単位課金，それぞれの特徴は以下のとおりである．

チャネル単位課金

-チャネル単位で課金を行う場合，U は一度だけ提案プロトコルを実行するだけで契約期間中はそのチャネルを視聴できるため効率が良い．

-各チャネルを構成する番組はあらかじめ BS によって決められているため，U は視聴したい番組を自由に選択することはできない．

番組単位課金

-番組単位で課金を行う場合，U は番組を契約するごとに提案プロトコルを実行しなければならず効率が悪い．

-U は，視聴したい番組を自由に選択できる．

3.2 安　全　性

提案方式における「ユーザのプライバシ」，「課金の正当性」，「プロトコルの健全性」，「ユーザの正当性」について考察する．

「ユーザのプライバシ」：復号鍵復元フェーズにおいて，ユーザ U は 1-out-of-n OT を用いて各分散情報データベース DB_i（$1 \leq i \leq n$）からチャネル cha_j（$1 \leq j \leq m$）の復号鍵 K_j の分散情報 K^i_j を取得するため，DB_i は自身が保持するいずれの分散情報を U が取得したかを知ることはできない．

n 個あるすべてのデータベースが結託した場合，DB_i は U の分散情報総取得回数 $NKA^u = NKA^u_{DB_1} + NKA^u_{DB_2} + \ldots + NKA^u_{DB_n}$ を知ることができる．また，DB_i は 1 つの分散情報の価格 y を知った場合，NKA^u と y から U の利用料金 $NKA^u y$ を知ることができる．しかし，$NKA^u y$ を構成するチャネルの組合せが複数存在する場合，$NKA^u y$ の内訳（U が視聴したチャネルの組合せ）を特定することはできない．そのような意味で，DB_i は $NKA^u y$ から U がいずれのチャネルを視聴したかを知ることはできない．放送局 BS についても同様のことがいえる．

BS が分散情報データベース $DB_1, DB_2, \ldots, DB_n$ を管理していた場合，U が同時にいくつの分散情報

データベースにアクセスしたかが分かってしまうため，BS は U がいずれのチャネルを視聴したかが分かってしまう．そのため，BS が $DB_1, DB_2, \ldots, DB_n$ を管理する場合には，U は視聴したいチャネルが複数あるときにはそれらの復号鍵の分散情報は一括して取得したり，復号鍵の分散情報を取得する際に分散情報データベースに同時にアクセスするのではなくランダム時間間隔でアクセスしたりするなどして，アクセスした分散情報データベースの個数から視聴したチャネルの特定を防がなければならない．

視聴フェーズにおいて，U は自身が視聴した cha_j の BS の署名付き視聴情報 $\sigma^{BS}_{view_j}$ を Mixnet を利用して BS へ送っているが，Mixnet が U に対して提供する匿名性より，BS は送られてきた $\sigma^{BS}_{view_j}$ から U を特定することはできない．

「課金の正当性」：請求フェーズにおいて，BS は各 DB_i からそれらが保持する U の分散情報取得回数 $NKA^u_{DB_i}$ と問合せに対する署名☆を取得し，$NKA^u_{DB_i}$ と U の問合せに対する署名の総数が等しくなることを確認する．このことにより，DB_i は自身が保持する $NKA^u_{DB_i}$ を偽ることはできない．したがって，BS は U に対して正しく利用料金を請求することができる．

「プロトコルの健全性」：セットアップフェーズにおいて，BS は，1回の分散情報の取得にかかる料金を y に設定し，K_j について $k_j y = x_j$ となるようにしきい値 k_j を求め，関数 *Share* を用いて n 個の分散情報を計算する．BS は，K_j，k_j，n を関数 *Share* に入力し，n 個の分散情報 $(K^1_j, K^2_j, \ldots, K^n_j) = Share(K_j, k_j, n)$ を得る．そして，BS は K^i_j（$1 \leq i \leq n$）が確かに K_j の分散情報であることを U が検証するために必要となる検証情報 v_{K_j} を公開する．このことから，BS が K_j について正しく分散情報を計算したのであれば，U は K^i_j（$1 \leq i \leq n$）が確かに K_j の分散情報であるかどうかを関数 *Verify* と BS が公開した K_j についての検証情報 v_{K_j} から検証することができる．したがって，U は $reject = Verify(K^i_j, j, v_{K_j})$ を得た場合，BS の不正を検出することができる．

視聴フェーズにおいて，Mixnet を利用することで BS は cha_j が視聴されたという情報である BS の署名付き視聴情報 $\sigma^{BS}_{view_j}$ を，それを BS へ送った U を知ることなしに得ることができる．

各チャネルの視聴情報には，BS の署名が付加されているため，BS の署名が付加されていない視聴情報は偽造されたものとして正当な視聴情報と区別することができる．このことから，偽造された視聴情報の送信による視聴率操作を防ぐことができる．

提案方式においては，同一の視聴情報を複数回送信するという不正を防ぐことはできない．しかし，提案方式においてタンパーフリーデバイス（*TFD*）を利用することでこの不正を防ぐことができる．*TFD* とは，*TFD* 内にある情報を知るために *TFD* を解析しようとすると，*TFD* 内にあるすべての情報が失われしまうという性質を持つ装置である．

以下に，U が cha_j を視聴したときに，*TFD* を利用し cha_j の視聴情報 $\sigma^{BS}_{view_j}$ の複数回送信を防止する方法を示す．BS と *TFD* は互いにマスタ鍵 MK を共有しているとし，*TFD* は共通鍵暗号アルゴリズムを利用できるとする．さらに，*TFD* はランダム鍵生成器 G_r を持っているとする．まず，U は復号した cha_j の復号鍵 K_j と暗号化されてブロードキャストされている cha_j とその視聴情報である c_j と $c\sigma^{BS}_{view_j}$ を *TFD* の入力とすることで，*TFD* は cha_j と $\sigma^{BS}_{view_j}$ を出力する．次に，*TFD* は G_r と乱数 r_j からランダム鍵 $RK_j = G_r(r_j)$ を生成し，RK_j と $\sigma^{BS}_{view_j}$ を共通鍵暗号アルゴリズムの暗号化アルゴリズム E の入力とし，$e\sigma^{BS}_{view_j} = E(RK_j, \sigma^{BS}_{view_j})$ を得る．さらに，*TFD* は MK と RK_j を E の入力とし $e'RK_j = E(MK, RK_j)$ を得る．*TFD* は，$e\sigma^{BS}_{view_j}$ と $e'RK_j$ を Mixnet を用いて BS へ一度だけ送信する．BS は，$mix(BS, e\sigma^{BS}_{view_j})$，$mix(BS, e'RK_j)$ により，$e\sigma^{BS}_{view_j}$ と $e'RK_j$ を得る．BS は，MK と $e'RK_j$ を共通鍵暗号アルゴリズムの復号アルゴリズム D の入力とし，$RK_j = D(MK, e'RK_j)$ を得る．最後に，BS は RK_j と $e\sigma^{BS}_{view_j}$ を D の入力とし，$\sigma^{BS}_{view_j} = D(RK_j, e\sigma^{BS}_{view_j})$ を得る．ここで，U が $\sigma^{BS}_{view_j}$ を再度送信しようとする場合を考える．U は，$\sigma^{BS}_{view_j}$ を再度送信するためには，マスタ鍵 MK を必要とする．しかし，U が MK を知るために *TFD* を解析した場合，*TFD* 内の情報である MK は失われてしまう．U は，MK を知ることができないため $\sigma^{BS}_{view_j}$ を再度送信できない．したがって，提案方式において *TFD* を利用することで同一の視聴情報の複数回送信を防ぐことができる．

「ユーザの正当性」：ユーザ U が，チャネル cha_j の復号鍵 K_j を正当な方法で取得したことを証明することを考える．このとき U は，任意の検証者に U が選択したデータベース $\{DB_{l1}, DB_{l2}, \ldots, DB_{lk_j}\} \subset \{DB_1, DB_2, \ldots, DB_n\}$ から送られてきた U の署名

☆ 問合せに対する署名は，$\sigma^u_{q_{j1}}, \sigma^u_{q_{j2}}, \ldots, \sigma^u_{q_{jNKA^u_{DB_i}}}$（$\{j1, j2, \ldots, jNKA^u_{DB_i}\} \subset \{1, 2, \ldots, m\}$）である．

$\sigma^{u}_{q_j}$ に対する各データベースの署名 $\sigma^{DB_{l1}}_{\sigma^{u}_{q_j}}, \sigma^{DB_{l2}}_{\sigma^{u}_{q_j}}, \ldots, \sigma^{DB_{lk_j}}_{\sigma^{u}_{q_j}}$ を示す．これにより，U は任意の検証者に対して K_j を $DB_{l1}, DB_{l2}, \ldots, DB_{lk_j}$ から取得した K_j の分散情報 $K_j^{l1}, K_j^{l2}, \ldots, K_j^{lk_j}$ から復元したということを証明することができる．

4. おわりに

本論文では，放送型コンテンツ配信におけるチャネル単位課金/番組単位課金の，ユーザのプライバシを保護した匿名視聴方式を提案した．提案方式により，ユーザが視聴したチャネルの情報をだれにも知られずに各チャネルを視聴することができる．また，放送局はユーザがいずれのチャネルを視聴したかを知ることなしにユーザに対して正当な利用料金を請求することができる．さらに放送局は，各チャネルに対して異なる価格を設定することができ，ユーザの協力によりユーザが視聴したチャネルの視聴情報をそのチャネルを視聴したユーザの情報を知ることなしに得ることができる．

参考文献

1) Rabin, M.: How to exchange secrets by oblivious transfer, Tech. memo TR-81, Aiken Computation Laboratory, Havard Univ. (1981).
2) Even, S., Goldreich, O. and Lempel, A.: A randomized protocol for signing contracts, *Comm. ACM*, Vol.28, No.6, pp.637–647 (1985).
3) Brassard, G., Crepeau, C. and Robert, J.-M.: All -or- nothing disclosure of secrets, *Advance in Cryptology, Proc. CRYPTO '86*, Vol.263 of Lecture Notes in Computer Science, pp.234–238 (1986).
4) Naor, M. and Pinkas, B.: Oblivious transfer and polynomial evaluation, *Proc. 31st Annual ACM Symposium on the Theory of Computing*, pp.245–254 (1999).
5) Naor, M. and Pinkas, B.: Oblivious transfer with adaptive queries, in: Advances in Cryptology, *Proc. CRYPTO '99*, Vol.1666 of Lecture Notes in Computer Science, pp.573–590 (1999).
6) Stern, J.P.: A New Efficient All-Or-Nothing Disclosure of Secrets Protocol, *ASIACRYPT*, pp.357–371 (1998).
7) Shamir, A.: How to share a secret, Comm. ACM, Vol.22, No.11, pp.612–613 (1979).
8) Blakley, G.R.: Safeguarding cryptographic keys, Proc. Nat. Computer Conf. AFIPS Conf., Vol.48, pp.313–317 (1979).
9) Pedersen, T.P.: Non-Interactive and Information Theoretic Secure Verifiable Secret Sharing, *Advances in Cryptology—CRYPTO '91*, LNCS 576, pp.129–140 (1992).
10) Feldman, P.: A Practical Scheme for Non-Interactive Verifiable Secret Sharing, *Proc. 28th FOCS*, pp.427–437 (1987).
11) Chaum, D.: Untraceable electronic mail, return addresses and digital pseudonyms, *Comm. ACM*, Vol.24, No.2, pp.84–88 (1981).
12) 千田浩司，小宮輝之，林 徹：匿名性確保と不正者追跡の両立が可能な通信方式，情報処理学会論文誌，Vol.45, No.8, pp.1873–188 (2004).
13) Fiat, A. and Naor, M.: Broadcast Encryption, *Advances in Cryptology—CRYPTO '93*, LNCS 773, pp.480–491 (1994).
14) 光成滋生，渡辺秀行，古田真紀，境 隆一，笠原正雄：楕円曲線上のペアリングを用いた不正者追跡法の拡張，コンピュータセキュリティ（CSEC），18-38, pp.261–266 (2002.7).
15) Aiello, B., Ishai, Y. and Reingold, O.: Priced Oblivious Transfer: How to Sell Digital Goods, Advances in Cryptology—EUROCRYPT 2001, Lecture Notes in Computer Science, Vol.2045, pp.119–135 (2001).
16) 飛田孝之，山本博紀，土井 洋，真島恵吾：利用履歴を秘匿できるコンテンツ配信・課金方式の改良，コンピュータセキュリティ（CSEC），pp.19–24 (2006.5.12).

(平成 18 年 11 月 27 日受付)
(平成 19 年 6 月 5 日採録)

櫻井 友二

平成 17 年東京電機大学工学部情報通信工学科卒業．平成 19 年東京電機大学大学院工学研究科情報通信工学専攻修士課程修了．同年株式会社日立製作所へ入社．

齊藤 泰一（正会員）

平成元年早稲田大学理工学部数学科卒業．平成 3 年早稲田大学大学院理工学研究科修士課程数学専攻修了．同年日本電信電話株式会社へ入社．平成 13 年中央大学理工学研究科情報工学専攻博士後期課程修了．平成 16 年より東京電機大学助教授（平成 19 年より准教授と名称変更）．暗号理論，情報セキュリティの研究に従事．博士（工学）．電子情報通信学会会員．