平成 12 年度

学士学位論文

モバイルコンテンツ流通プロトコルの検討

Protocols of Contents Communication on Mobile

1010433　　林 竜也

指導教員　　清水 明宏

２００１ 年 ２ 月 ５ 日

高知工科大学 情報システム工学科

要 旨

モバイルコンテンツ流通プロトコルの検討

林 竜也

モバイル環境において，インターネット上でコンテンツ提供者が，コンテンツ購入の資格を有するユーザに対し，安全，且つ効率的にデジタルコンテンツを提供する方式を提案する．ユーザはプリペイドカードを購入することで，コンテンツ購入資格を取得し，ユーザの認証には SAS-K 認証方式を適用する．そしてコンテンツ配信の信頼性と，認証，課金を行うブローカの処理量を考慮に入れ，コンテンツ要求から，認証，課金，コンテンツ配信までの手順を定めたプロトコルを２つ提案した．１つは，コンテンツ配信において高い信頼性を得ることができるプロトコル．もう一つは，大規模なユーザに対して安定したコミュニケーションを提供できるプロトコルである．

キーワード　モバイル，小額課金，デジタルコンテンツ，SAS-K 認証方式

Abstract

Protocols of Contents Communication on Mobile

Contents delivery services, which enable contents providers to securely deliver digital contents via the internet, are presented on Mobile. Funds in the user's account can be plenished from purchasing the prepaid card,and SAS-K is applied to the user authentication. I presented two protocols, which stipulates data flow from contents riquest to contents provide, considering dependable contents offer and the burden of borker. One can communicate very dependable contents offer, and other can present stability comunication on large-scale.

key words Mobile Communication, Micropayment, Digital Contents, SAS-K

目次

第 1 章　はじめに　1

1.1　モバイル EC 市場の形成 1

1.2　モバイルコンテンツ流通の現状 2

1.3　研究目的 3

第 2 章　小額課金システム　4

2.1　Broker 4

2.2　課金手段 5

2.2.1　クレジットカード決済の問題点 5

2.2.2　電子マネー決済の問題点 5

2.2.3　プリペイド決済の適用 6

2.3　プリペイド課金への SAS-K 認証方式の適用 6

2.4　小額課金システムの概要 7

記号の表記 7

2.4.1　プリペイド型課金サービスの初期登録方法 8

(1) プリペイドカードの購入 8

(2) 課金センタの予備登録 8

(3) ユーザ側の登録 8

(4) 課金センタ側の登録 9

2.4.2　プリペイド型課金サービスの認証方法 11

(1) 被認証側の認証データ生成 11

(2) 認証側の認証アルゴリズム 11

(3) 認証完了後の処理 12

第 3 章　コンテンツ流通プロトコルの提案　14

3.1　コンテンツ流通モデル 14

記号の表記 15

3.2　Protocol.1 16

3.2.1　コンテンツリクエスト手順 16

3.2.2　認証手順 17

Vendor 認証 17

User 認証 17

3.2.3　課金手順 18

3.2.4　コンテンツ送信手順 18

3.2.5　エラー処理 20

認証手順において Vendor が認証されず，User のみが認証された場合 20

認証手順において Vendor のみが認証され，User が認証されたなかった場合 20

認証手順において Vendor も User も認証されなかった場合 20

課金手順において，コンテンツ金額 Ccharge がユーザの残高 M を上回る場合， 20

3.3　Protocol.2 21

3.3.1　コンテンツリクエスト手順 21

3.3.2　認証手順 21

3.3.3　コンテンツ送信手順 22

3.3.4　課金手順 22

3.3.5　エラー処理 24

認証手順において User が認証されなかった場合 24

コンテンツ送信手順において，コンテンツ金額 C_{CHARGE} がユーザの残高 M_{USER} を上回る場合 24

目次

第 4 章　プロトコルの考察 **25**

4.1　コンテンツ配信における信頼性 25

4.1.1　Protocol.1 25

4.1.2　Protocol.2 25

4.2　Broker の負荷 26

4.2.1　Protocol.1 26

4.2.2　Protocol.2 26

4.3　プロトコルの適用領域 27

第 5 章　おわりに **28**

謝辞 **29**

参考文献 **30**

付録 A　FEAL-8 プログラム **31**

第 1 章

はじめに

1.1 モバイル EC 市場の形成

1999 年 2 月に携帯電話端末からインターネットにアクセスできるサービスが開始されてから，ブラウジングできる電話「ブラウザフォン」が爆発的に普及してきた．現在の PHS を含む携帯電話の加入者数は 6400 万人で、そのうち約 46%，実に 3000 万人がインターネットアクセスのサービスを利用している．これは、現在パソコンからインターネットを利用している人口とほぼ同数である．[1]

かつてインターネットの普及に伴い，新しい流通経路としてインターネットが注目され，それまで電話や手紙を利用して行われてきた通信販売が電子商取引（Electronic Commerce : EC）というネットワークを介した形態をとるようになってきた．その際の課金に，クレジットカードによる決済をインターネット上で行う仕組や，またはネットワーク上であたかも現金のように価値そのものが流通する電子マネーの研究がなされ，現在の EC 市場が築かれてきた．

そして今，携帯電話のようなポケットに納まる端末からインターネットにアクセスできるようになり，携帯電話からのインターネットアクセスサービスの利用者数も今やパソコンのインターネット利用者数に迫る勢いで増加してきた．各コンテンツベンダはこれを新たなビジネスチャンスと捉え，ビジネスモデルを模索している．

1.2　モバイルコンテンツ流通の現状

携帯電話でのモバイルコンテンツの課金では，携帯電話キャリアによる料金徴収代行システムという仕組みがある．キャリアメニューに入ると，ユーザの認証，請求書の発行，未回収の売掛金の処理など，一番頭を悩ませる情報量の回収を全てキャリアが受け持ってくれるというメリットがあり，コンテンツ配信から料金回収までを含めたビジネスモデルが分かりやすい．[2](図 1.1)

しかしキャリアが積極的にサイトを公募しているわけではないし，キャリアメニュー参入の際の審査もハードルが高く，時間もかかることから，ベンチャー企業などのコンテンツベンダのキャリアメニュー参入は困難である．そのため，キャリアメニューに参入できず独立系サイト向けの検索エンジンに登録しているコンテンツベンダは，コンテンツ配信の確実なビジネスモデルが見出せず，ボランティア運営を強いられているのが現状である．

現在最も活況を呈している NTT ドコモの i-mode サービスでは，ドコモが 9%の情報料回収料を徴収している公式サイトは約 1,360 なのに対し，それ以外の情報料の発生してない無料サイトの数は約 32,440 にもなる．

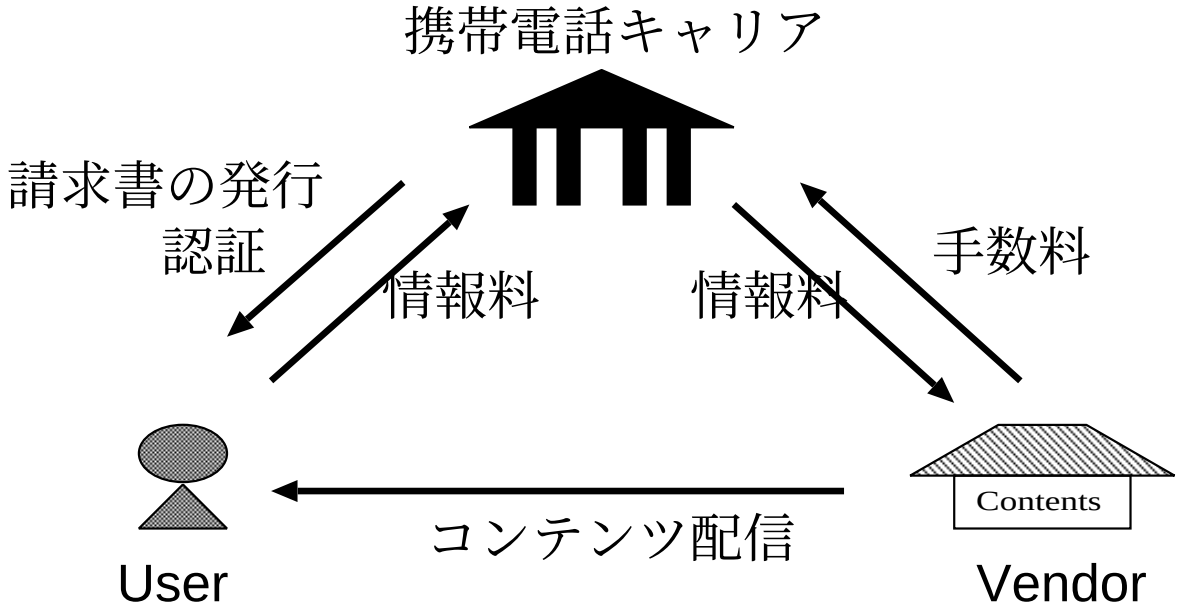


図 1.1　キャリアによるコンテンツ流通モデル

1.3　研究目的

携帯電話キャリアに参入しているコンテンツベンダに比べ，圧倒的に多い独立系のコンテンツベンダのビジネスとして成立できない最大の理由であるコンテンツに対する課金システムを構築することで，これらの独立系コンテンツベンダにも新たなビジネスチャンスが生まれ，モバイル EC 市場は更に拡大，発展していくと考えられる．

本研究では，モバイル環境で，キャリアに依存せずコンテンツ料金を徴収するシステムを検討する．

第 2 章

小額課金システム

2.1 Broker

モバイル環境におけるコンテンツ課金を行うとき，コンテンツを利用，購入する User と，コンテンツを提供する Vendor との間に，キャリアの代わりに User(Vendor) の認証や課金を行う仲介役 (以下 Broker) が必要である．(図 2.1)

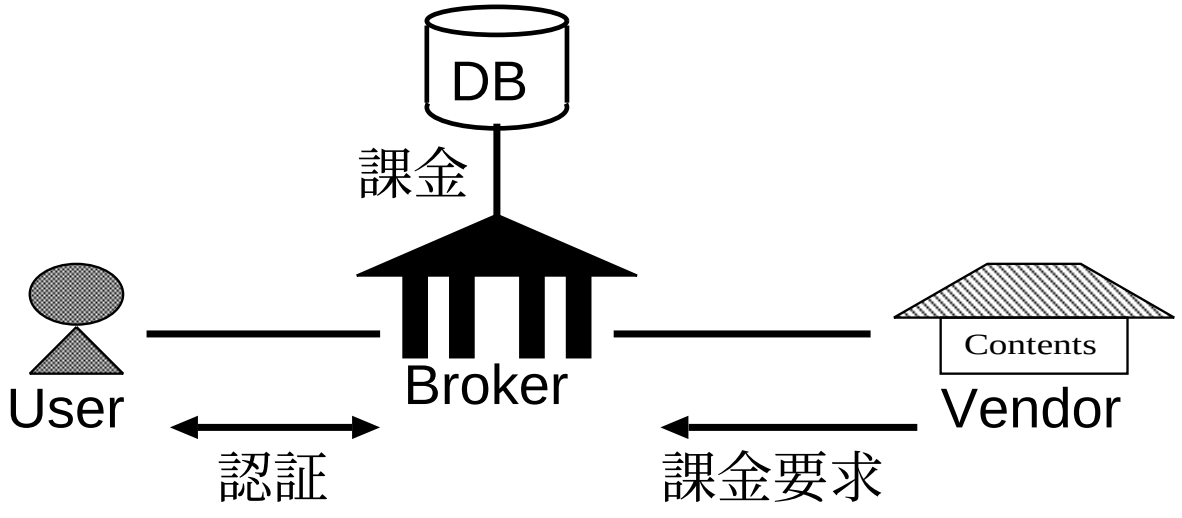


図 2.1　Broker の位置付け

この章では，モバイル環境における小額課金システムに適した Broker の課金手段と認証方法について述べる．

2.2　課金手段

現在，電子商取引に適用されている課金方法には，クレジットカードによる決済，電子マネーによる決済，プリペイドカードによる決済，がある．

2.2.1　クレジットカード決済の問題点

モバイル端末におけるコンテンツ流通では，モバイルならではの「いつでも，どこでも」インターネットにアクセスできるといった特性から，物品の取引より 画面メモ等の画像データや着信メロディ等の音楽データ，占い等のテキストデータといったデジタルコンテンツが主流となってきている．それら一つ一つの利用料金は，物品購入料金に比べ遥かに小額である．

インターネット上の電子商取引で最も普及しているクレジットカードによる決済では，課金の際に加盟店に対してカード番号が無効でないか，利用限度額をこえてないかなどの与信照会という処理が必要になる．[3] クレジットカード決済は，このコンテンツ料金の課金に対する与信照会の手数料が，コンテンツ料金そのものに対して甚大になってしまうため，小額課金に適用する決済手段として適当でないと言える．

さらに携帯電話からインターネットにアクセスし，デジタルコンテンツを利用する人々は，クレジットカード等を持たない 10・20 代に最も多いという現状である．

2.2.2　電子マネー決済の問題点

電子マネーとは，インターネット上で現金に代わり，コンテンツ購入料金に支払われたり，他人への権利の譲渡が可能な，仮想の現金を流通させるシステムである．コンテンツ流通の課金に適用する際，クレジットカードに比べてよりコストを安くできるが，高度なセキュリティが要求されるため，どこまでコストを削減できるかという課題がある．このセキュリティの要求のため，モバイル環境においては，端末側での処理量にも問題がでてくる．[4]

2.2.3　プリペイド決済の適用

プリペイドカードによる決済では，あらかじめプリペイドカードを購入する時に，Beoker が実決済をまとめて済ませることで，取引 1 件当たりの決済コストを削減できる．実決済を済ませ，実世界の金銭価値をインターネットの金銭価値に置き換えた後は，コンテンツ購入の際に Broker による簡易な与信が行われるだけであり，運用コストは安くつく．

これらの理由から，本課金システムでは，当研究室で研究を進めているプリペイド型の課金システム [5] を採用する．

2.3　プリペイド課金への SAS-K 認証方式の適用

インターネット上でのユーザ認証は，ネットワークを流れる認証データが第三者により，簡単に盗聴，改竄される恐れがある．更に，モバイル端末におけるコンテンツ流通プロトコルの検討という事で，ユーザ認証の際の認証方式の適用には，モバイル端末，特に携帯端末のような処理能力の低い端末への実装ということを考慮に入れなければならい．

SAS-K 認証方式 [6] はワンタイムパスワード認証方式で，ネットワークを流れる認証データが毎回異なる為，第三者からの盗聴，なりすまし，データの改竄に強固な認証方式である．更に，ユーザ端末における 1 回の認証に必要な処理は，5 回のハッシュ演算と 3 回のビット排他的論理和演算のみで，端末に必要な記憶領域はユーザ ID データと 1 つの乱数のみである．これは，処理能力の低いモバイル端末にも簡易に実装が可能である．

これらの理由により処理能力の低いモバイル端末で十分なセキュリティを保証できる認証方式として SAS-K 認証方式を適用する．

なお，本研究ではハッシュ演算に FEAL-8(付録．A) を用いる

2.4　小額課金システムの概要

ここでは，SAS-K 認証方式をプリペイド型小額課金システムに適用する際の，初期データの登録方法と，n 回目の認証方法を述べる．

記号の表記

データ記号	データ長	用途
P	16byte	P_{INDEX} と P_{MASK} からなるプリペイドカード番号.
P_{INDEX}	8byte	外部に洩れても支障のない，ユーザを判断するインデックスデータ
P_{MASK}	8byte	外部から参照不可能な，ユーザ ID データ A と初期認証データ E_0^2 へのマスク.
M	8byte	プリペイド番号 P の購入額．初期登録時にユーザ残高 M_{USER} として登録される.
M_{USER}	8byte	ユーザの残高.
A	8byte	ハッシュ関数のキーとなるユーザ ID.
S	8byte	外部から参照不能のユーザパスワード.
N_n	8byte	n 回目の認証時に使用される乱数
E_n^m	8byte	n 回目認証時において，ユーザ ID データ A をキーにユーザパスワード S と乱数 N_n の排他的論理和 S$\oplus N_n$ に一方向性関数を m 回適用したもの. E_n^1=E(A,S$\oplus N_n$) E_n^2=E(A,E_n^1)

2.4.1　プリペイド型課金サービスの初期登録方法

SAS-K 認証方式を小額課金のユーザ認証に利用するにあたり，ユーザはユーザインデックス P_{INDEX} と初期認証データであるユーザ ID データ A と認証データ E_0^2 をセキュアなルートで課金センタに登録せねばならない.
ここではユーザ ID データ A と認証データ E_0^2 のセキュアな登録方法を示す．(図 2.2)

(1) プリペイドカードの購入

ユーザはコンビニエンスストア等，外部から盗聴不可能な専用線を課金センタと接続している機関にて，インデックスとして機能するインデックス部 P_{INDEX} と，外部から認識できないマスキング部 P_{MASK} から構成されたプリペイド番号 P を購入し，ユーザ端末にプリペイド番号 P を登録する.

(2) 課金センタの予備登録

コンビニエンスストアはユーザがプリペイド番号 P を購入した直後に，盗聴される恐れのない専用線にてプリペイド番号 P とプリペイド番号 P の購入額 M を課金センタへ送信する.

(3) ユーザ側の登録

ユーザはプリペイド番号 P が登録されたユーザ端末を用いて，ユーザ ID データ A と，ユーザパスワード S と乱数 N_0 との排他的論理和 S$\oplus N_0$ を基に初期認証データ E_0^2 を生成し，初期認証データ E_0^2 とプリペイド番号 P のマスキング部 P_{MASK} との排他的論理和 X を算出し，ユーザ ID データ A とインデックス部 P_{INDEX} と前

記排他的論理和 X を課金センタに送信し, ユーザ ID データ A と乱数 N_0 を端末に記憶する.

(4) 課金センタ側の登録

課金センタではユーザ ID データ A とインデックス部 P_{INDEX} と前記排他的論理和 X を受信し，インデックス部 P_{INDEX} をインデックスとしてプリペイド番号 P を検索しマスキング部 P_{MASK} を取得する．そしてマスキング部 P_{MASK} と前記排他的論理和 X との排他的論理和演算から初期認証データ E_0^2 を復元し，復元された初期認証データ E_0^2 とユーザ ID データ A とユーザインデックス P_{INDEX} を登録する．そしてプリペイド番号 P の購入金額 M をユーザ残高 M_{USER} として登録する.

2.4　小額課金システムの概要

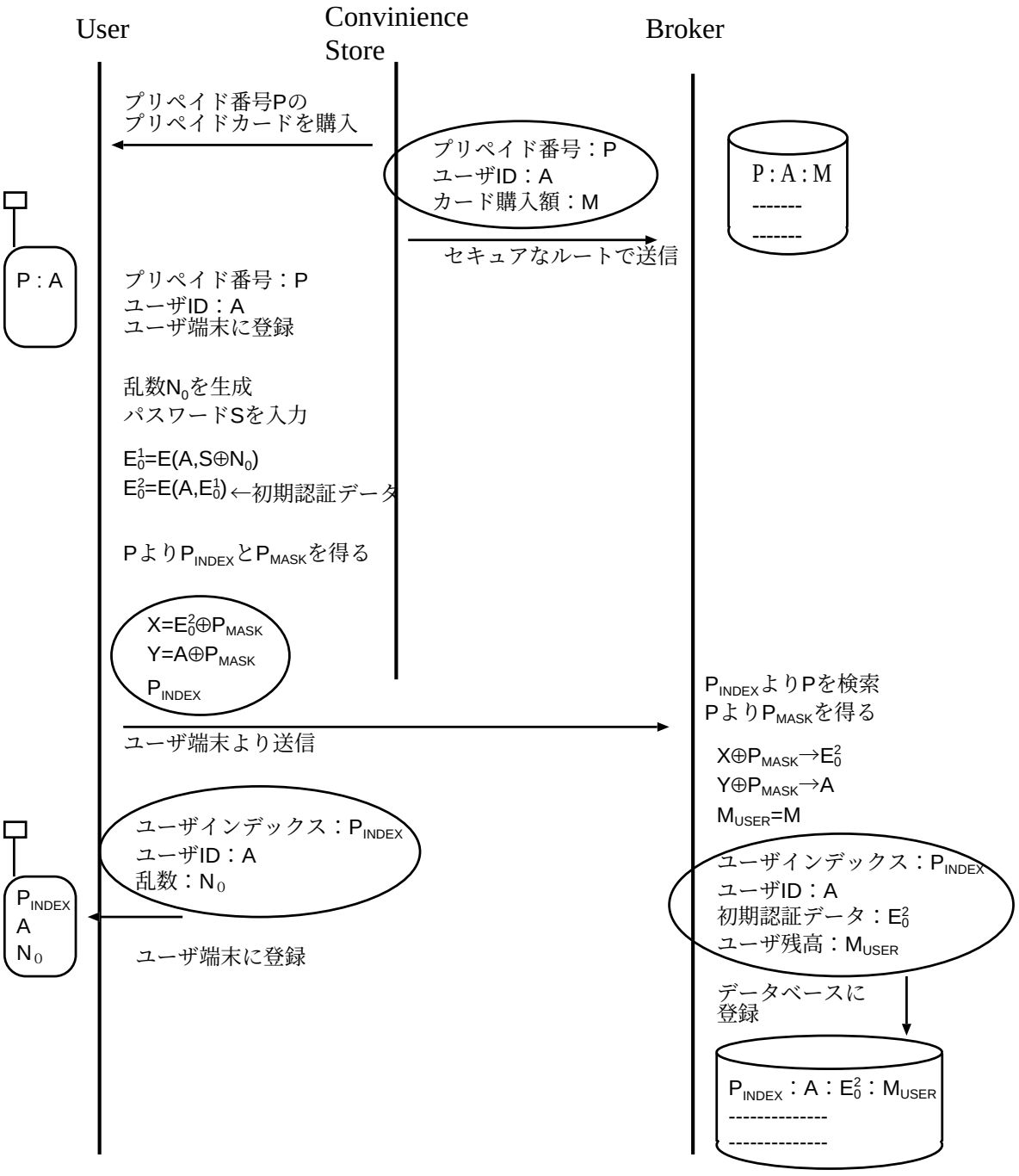


図 2.2　初期登録手順

2.4.2　プリペイド型課金サービスの認証方法

プリペイド型課金サービスの認証方法には SAS-K 認証方式を適用する．
以下に SAS-K 認証方式の n 回目の認証手順を示す．(図 3.1)

(1) 被認証側の認証データ生成

まず被認証者側で，n-1 回目に登録した n 回目の認証用の乱数 N_n とユーザ ID データ A を呼び出し，ユーザ ID データ A をキーにユーザが入力したパスワード S と乱数 N_n の排他的論理和 S$\oplus N_n$ を一方向性関数 E で暗号化し E_n^1 を得る．さらにユーザ ID データ A をキーに，E_n^1 を一方向性関数 E で暗号化し E_n^2 を得る．

次に次回認証用に乱数 N_{n+1} を生成し，そしてユーザ ID データ A をキーに，ユーザが入力したパスワード S と乱数 N_{n+1} の排他的論理和 S$\oplus N_{n+1}$ を一方向性関数 E で暗号化し E_{n+1}^1 を得る．さらにユーザ ID データ A をキーに，E_{n+1}^1 を一方向性関数 E で暗号化し E_{n+1}^2 を得る．さらにユーザ ID データ A をキーに，E_{n+1}^2 を一方向性関数 E で暗号化し E_{n+1}^3 を得る．

得られたデータのうち，E_n^1 と E_n^2 と E_{n+1}^3 の排他的論理和演算により今回のワンタイム認証データ $E_n^1 \oplus E_n^2 \oplus E_{n+1}^3$ を，E_{n+1}^2 と E_n^2 の排他的論理和演算により次回のワンタイム認証データ $E_{n+1}^2 \oplus E_n^2$ を算出し，ユーザインデックス P_{INDEX} とともに認証側に送信する．

(2) 認証側の認証アルゴリズム

今回のワンタイム認証データ $E_n^1 \oplus E_n^2 \oplus E_{n+1}^3$ と次回のワンタイム認証データ $E_{n+1}^2 \oplus E_n^2$ とユーザインデックス P_{INDEX} を受信した認証側は，まず登録されている認証データ E_n^2 を呼び出し，次回のワンタイム認証データ $E_{n+1}^2 \oplus E_n^2$ と認証データ E_n^2 の排他的論理和演算により E_{n+1}^2 を復元する．

次に，ユーザインデックス P_{INDEX} からユーザ ID データ A を検出し，ユーザ ID データ A をキーに復元された E^2_{n+1} を一方向性関数 E で暗号化し E^3_{n+1} を得る．

そして今回のワンタイム認証データ $E^1_n \oplus E^2_n \oplus E^3_{n+1}$ と E^3_{n+1} の排他的論理和演算により $E^1_n \oplus E^2_n$ を復元し，さらに復元された $E^1_n \oplus E^2_n$ と認証データ E^2_n の排他的論理和演算により E^1_n を復元する．

ユーザ ID データ A をキーに復元された E^1_n を一方向性関数 E で暗号化し E^2_n を得て，得られた E^2_n を認証データ E^2_n と比較し，同一のデータであることが認められた時認証が成立する．

(3) 認証完了後の処理

認証が完了すると，認証側では認証データ E^2_n を破棄し，次回認証用に E^2_{n+1} を登録する．

被認証側では乱数 N_n を破棄し，次回認証データ生成用に乱数 N_{n+1} を登録する．

2.4　小額課金システムの概要

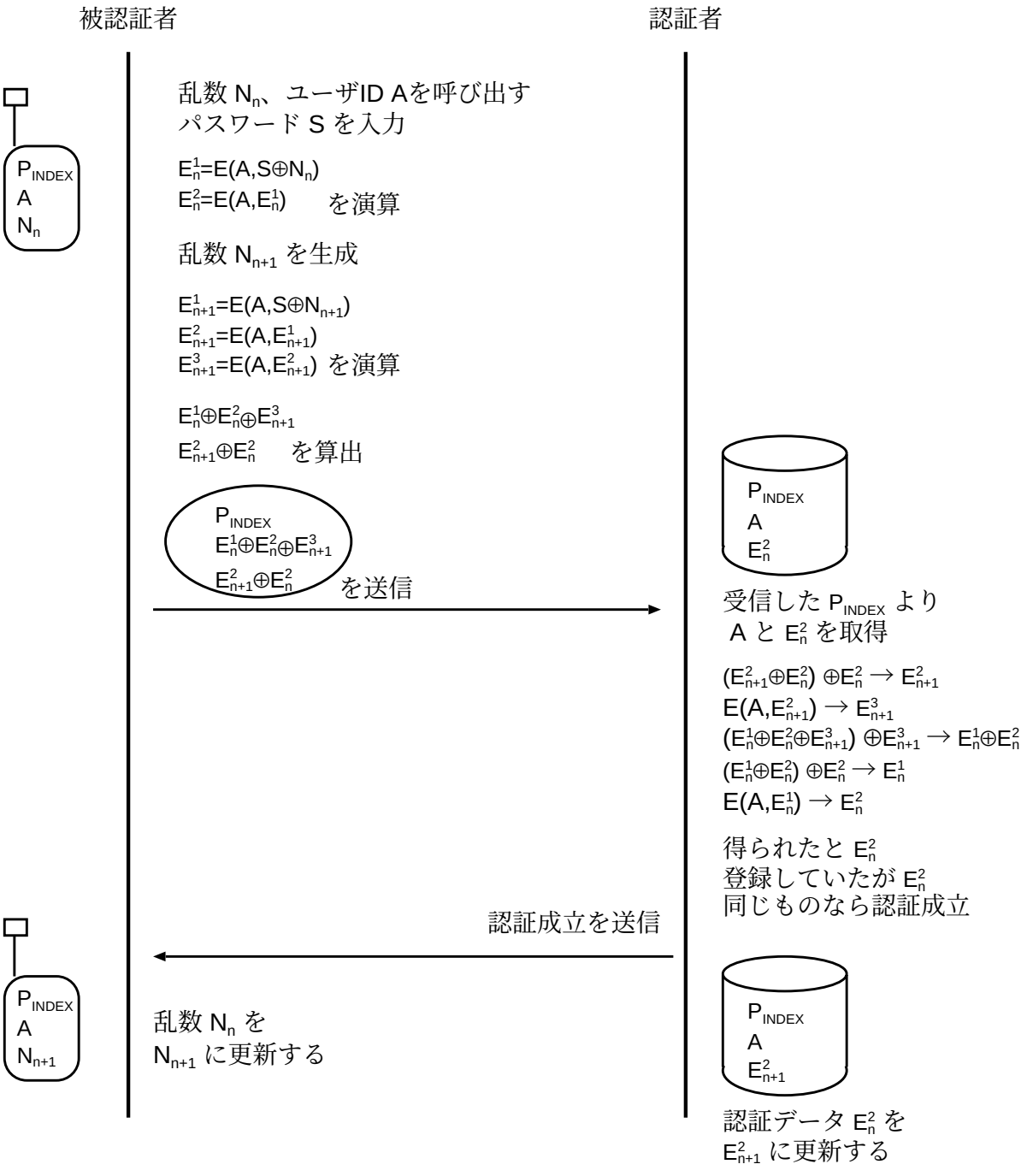


図 2.3　認証手順

第 3 章

コンテンツ流通プロトコルの提案

3.1　コンテンツ流通モデル

モバイル環境におけるコンテンツ課金は，コンテンツを利用，購入する User と，コンテンツを提供する Vendor と，User(Vendor) の認証と課金を行う Broker の 3 者間のデータのやりとりで構成される．(図**??**)

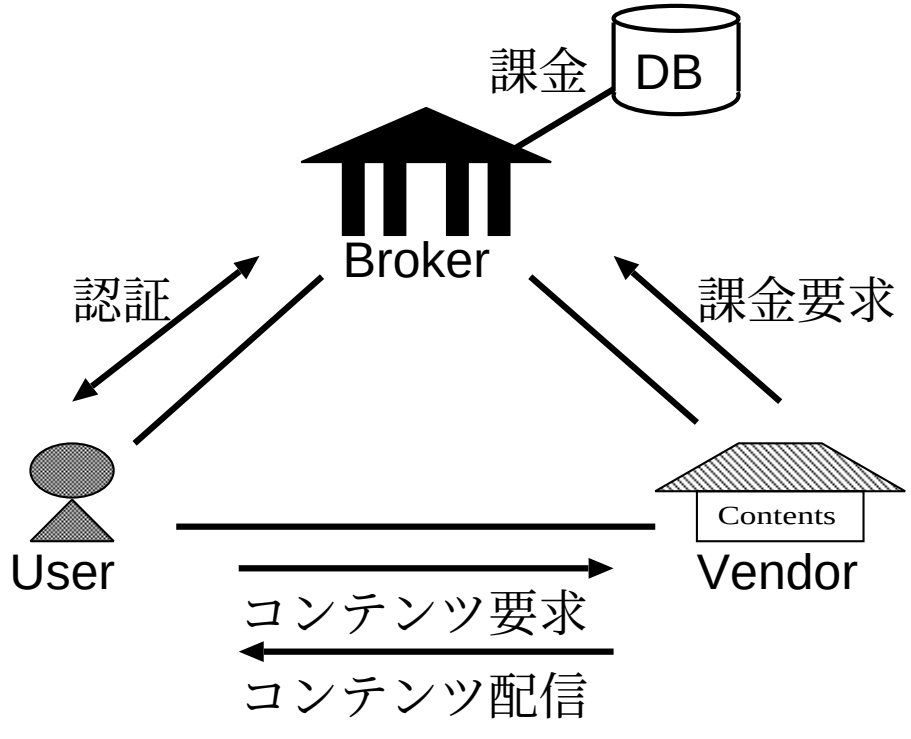


図 3.1　コンテンツ流通モデル

記号の表記

課金，認証において，前章で挙げた記号の他に，コンテンツを表すため以下の記号を使用する．

データ記号	データ長	用途
C_{ID}	8byte	コンテンツごとに割り振られる ID
C_{URL}	128byte	コンテンツの存在する URL データ
CONTENTS	任意	画像，音源，テキストなどのコンテンツデータ
C_{CHARGE}	8byte	コンテンツの料金
M_{USER}	8byte	ユーザの残高

また，Protocol.1 では，User と Vendor にそれぞれ認証が必要になってくるので，認証データ E_n^m をはじめ，認証に関わるデータを以下のように区別する．

データ記号	データ長	用途
Pu_{INDEX}	8byte	ユーザインデックスデータ
A	8byte	ユーザ ID データ
$U_n^1 \oplus U_n^2 \oplus U_{n+1}^3$	8byte	今回ユーザ認証用データ
$U_{n+1}^2 \oplus U_n^2$	8byte	次回ユーザ認証用データ
Pv_{INDEX}	8byte	ベンダインデックスデータ
B	8byte	ベンダ ID データ
$V_n^1 \oplus V_n^2 \oplus V_{n+1}^3$	8byte	今回ベンダ認証用データ
$V_{n+1}^2 \oplus V_n^2$	8byte	次回ベンダ認証用データ

前章では，User，Broker 間の課金方式と認証方式を述べたが，本章では User，Vendor，Broker 3 者間における，実際にコンテンツリクエストが発生してから認証，課金，コンテンツ配信までのフロー (手順) を定めたプロトコルについて述べる．

今回は，User，Broker，Vendor の処理量，User 数に伴う Broker への負荷，コンテンツ配信の信頼性を考慮し，そのプロトコルの適用領域により，異なる 2 つのプロトコルを考案した．

3.2 Protocol.1

このプロトコルでは，コンテンツ配信の信頼性に徹底的に重点を置き，それに伴う User，Broker の負荷も考慮に入れ設計した．

コンテンツ配信の信頼性向上のため，User だけでなく Vendor も認証を必要とし，コンテンツも Broker を中継して配信される．Broker に処理が集中しないように，User，Vendor の認証の際のデータのやりとりを Vendor とのみ行い，User から Broker に問い合わせるのはエラーが発生した時のみである．

3.2.1 コンテンツリクエスト手順

コンテンツ利用希望ユーザ（以下 User）はコンテンツプロバイダ（以下 Vendor）にユーザインデックス Pu_{INDEX} と，今回ユーザ認証用データ $U_n^1 \oplus U_n^2 \oplus U_{n+1}^3$ と，次回ユーザ認証用データ $U_{n+1}^2 \oplus U_n^2$ と, 要求コンテンツのコンテンツ ID データ C_{ID} を送信する．

ユーザインデックス Pu_{INDEX} と，今回ユーザ認証用データ $U_n^1 \oplus U_n^2 \oplus U_{n+1}^3$ と，次回ユーザ認証用データ $U_{n+1}^2 \oplus U_n^2$ と要求コンテンツのコンテンツ ID データ C_{ID} を受信した Vendor は，ユーザインデックス Pu_{INDEX} と，今回ユーザ認証用データ $U_n^1 \oplus U_n^2 \oplus U_{n+1}^3$ と，次回ユーザ認証用データ $U_{n+1}^2 \oplus U_n^2$ と，要求コンテンツのコンテンツ ID データ C_{ID} と，プロバイダインデックス Pv_{INDEX} と，今回プロバイダ認証用データ $V_n^1 \oplus V_n^2 \oplus V_{n+1}^3$

と, 次回プロバイダ認証用データ $V_{n+1}^2 \oplus V_n^2$ とコンテンツ URL データ C_{URL} とプロバイダ ID データ B を基に生成された Bx の排他的論理和 $C_{URL} \oplus$Bx と，プロバイダ ID データ V を課金センタ（以下 Broker）に送信する.

3.2.2 認証手順

ユーザインデックス Pu_{INDEX} と，今回ユーザ認証用データ $U_n^1 \oplus U_n^2 \oplus U_{n+1}^3$ と，次回ユーザ認証用データ $U_{n+1}^2 \oplus U_n^2$ と，要求コンテンツのコンテンツ ID データ C_{ID} と, ベンダインデックス Pv_{INDEX} と，今回ベンダ認証用データ $V_n^1 \oplus V_n^2 \oplus V_{n+1}^3$ と, 次回ベンダ認証用データ $V_{n+1}^2 \oplus V_n^2$ とコンテンツ URL データ C_{URL} とベンダ ID データ B を基に生成された Bx の排他的論理和 $C_{URL} \oplus$Bx と，ベンダ ID データ V を受信した Broker は，Vendor と User の認証を行う.

Vendor 認証

ベンダインデックス Pv_{INDEX} よりベンダ ID データ B を検出し，今回ベンダ認証用データ $V_n^1 \oplus V_n^2 \oplus V_{n+1}^3$ と，次回ベンダ認証用データ $V_{n+1}^2 \oplus V_n^2$ を用いて SAS-K 認証方式によるベンダ認証を行う.

User 認証

ユーザインデックス Pu_{INDEX} よりユーザ ID データ A を検出し，今回ユーザ認証用データ $U_n^1 \oplus U_n^2 \oplus U_{n+1}^3$ と，次回ユーザ認証用データ $U_{n+1}^2 \oplus U_n^2$ を用いて SAS-K 認証方式によるユーザ認証を行う.

3.2.3 課金手順

User も Vendor も Broker に登録されている正当な要求者と認証された場合，ベンダ ID データ B とコンテンツ ID データ C_{ID} からコンテンツ金額 C_{CHARGE} を取得し，コンテンツ金額 C_{CHARGE} がユーザの残高 M_{USER} を上回らない時，今回残高 M_{USER} とコンテンツ金額 C_{CHARGE} の差額を残高 M_{USER} として登録し，E^2_{n+1} を次回ユーザ認証用データとして登録する．

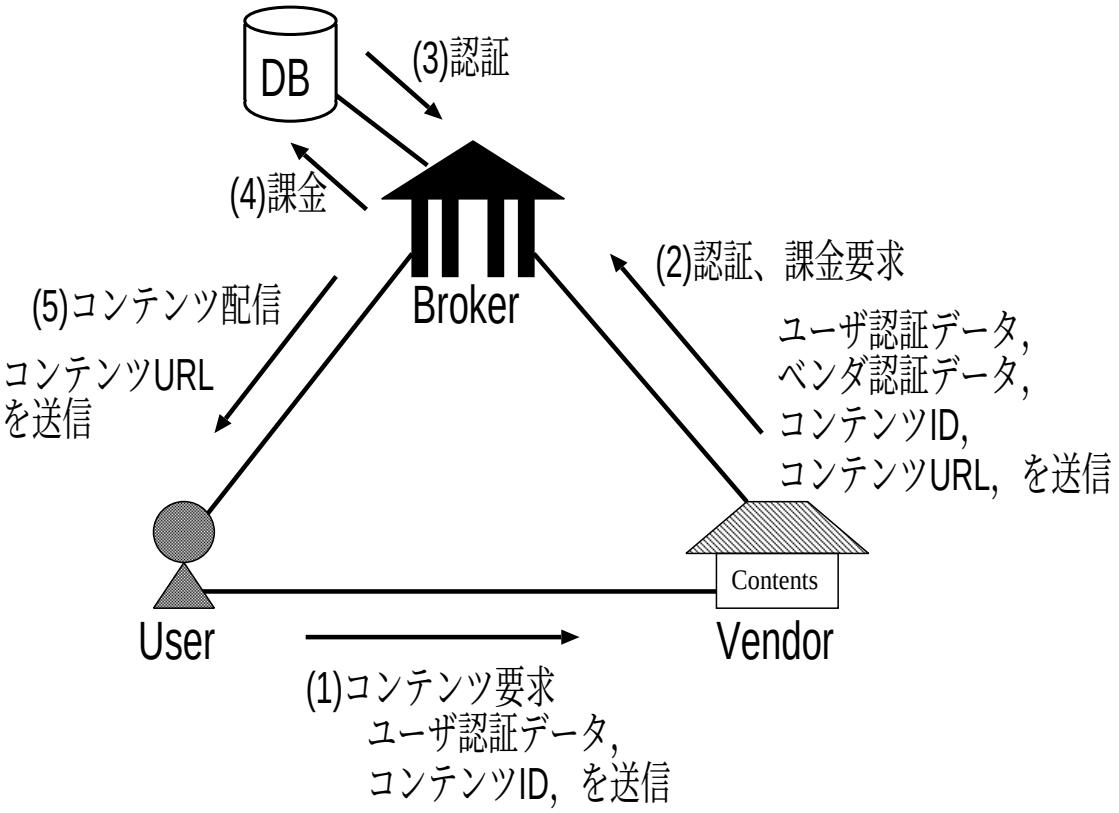
3.2.4 コンテンツ送信手順

前記排他的論理和 $C_{URL}\oplus$Bx とベンダ ID データ B を基に生成された Bx との排他的論理和演算により算出した C_{URL} と，ユーザ認証データ U^2_n を基に生成された U^2_nx の排他的論理和演算により算出された $C_{URL}\oplus U^2_n$x を User へ送信する．

前記排他的論理和 $C_{URL}\oplus U^2_n$x を受信した User は，ユーザ端末に登録されているユーザ認証データ U^2_n を基に U^2_nx を生成し，前記排他的論理和 $C_{URL}\oplus U^2_n$x と U^2_nx の排他的論理和演算により C_{URL} を復元し，C_{URL} を URL に入力し要求コンテンツを取得する．

要求コンテンツを取得後，先程次回ユーザ認証用データ算出に用いた乱数 Nn+を登録する．

Protocol.1



(1) User は Vendor にユーザ認証データとコンテンツIDを送信する.

(2) Userからコンテンツ要求を受信した Vendor は,
受信したデータに, ベンダ認証データとコンテンツURLを付属させ,
Broker に送信する.

(3) Broker はVendorとUser の認証を行う.

(4) 認証が成立すれば, User, Vendorに対して課金を行う.

(5) Userにコンテンツ URLを送信する

図 3.2 Prtotocol.1

3.2.5 エラー処理

認証手順において Vendor が認証されず，User のみが認証された場合

Broker は Vendor に対して「認証失敗」メッセージを送信し，ユーザインデックスデータを元に User に「認証データ更新」と「再度認証要求」のメッセージを送信する．

User は次回ユーザ認証用データ算出に用いた乱数 N_{n+1} を登録し，コンテンツ要求手順に戻る．

認証手順において Vendor のみが認証され，User が認証されたなかった場合

Broker は Vendor に対して「ユーザ認証失敗」のメッセージを送信する．

Vendor は User に「認証失敗」と「パスワード確認」のメッセージを送信する．

User はコンテンツ要求手順に戻る．

認証手順において Vendor も User も認証されなかった場合

Broker は Vendor に対して「認証失敗」メッセージを送信し，ユーザ ID が存在するならば User に「認証データ更新要求」を警告する．

課金手順において，コンテンツ金額 Ccharge がユーザの残高 M を上回る場合，

Broker は Vendor に対して「残高不足」のメッセージを送信し，ユーザ認証データを次回ユーザ認証データに書き換える．

Vendor は User に「残高不足」のメッセージを送信する．

User は次回ユーザ認証用データ算出に用いた乱数 N_{n+1} を登録する．

3.3 Protocol.2

このプロトコルでは，Broker の処理負担に重点を置き，User 数の増加にも安定して対応できるよう設計した．

Broker は Vendor の認証を行わず，User の要求をもとに Vendor へコンテンツ配信要求を通知する．Broker は主に User とデータのやりとりを行う．

また，コンテンツは Broker を中継しないので，画像データや音源データなど，様々な形態をとることが可能である．

3.3.1 コンテンツリクエスト手順

コンテンツ利用希望ユーザ（以下 User）は，要求コンテンツを所持しているコンテンツプロバイダ (以下 Vendor) より，ベンダ ID データ V と，要求コンテンツのコンテンツ ID データ C_{ID} を得て，課金センタ (以下 Broker) にユーザインデックス Pu_{INDEX} と，今回ユーザ認証用データ $E_n^1 \oplus E_n^2 \oplus E_{n+1}^3$ と，次回ユーザ認証用データ $E_{n+1}^2 \oplus E_n^2$ と，ベンダ ID データ V と，要求コンテンツのコンテンツ ID データ C_{ID} を送信する．

3.3.2 認証手順

ユーザインデックス Pu_{INDEX} と，今回ユーザ認証用データ $E_n^1 \oplus E_n^2 \oplus E_{n+1}^3$ と，次回ユーザ認証用データ $E_{n+1}^2 \oplus E_n^2$ と，ベンダ ID データ V と，要求コンテンツのコンテンツ ID データ C_{ID} を受信した Broker は，ユーザインデックス Pu_{INDEX} よりユーザ ID データ A を検出し，今回ユーザ認証用データ $E_n^1 \oplus E_n^2 \oplus E_{n+1}^3$ と，次回ユーザ認証用データ $E_{n+1}^2 \oplus E_n^2$ を用いて SAS-K 認証方式によるユーザ認証を行う．

3.3.3 コンテンツ送信手順

User が正当な要求者として認証された時，Broker はベンダ ID データ V より Vendor を割出し，要求コンテンツ ID データ C_{ID} に相当するコンテンツ ID とコンテンツ金額 C_{CHARGE} を確認する．Broker は，コンテンツ金額 C_{CHARGE} がユーザの残高 M_{USER} を上回らない時，Vendor に対して要求コンテンツ ID データ C_{ID} とともに，User へのコンテンツ配信要求を送信する．

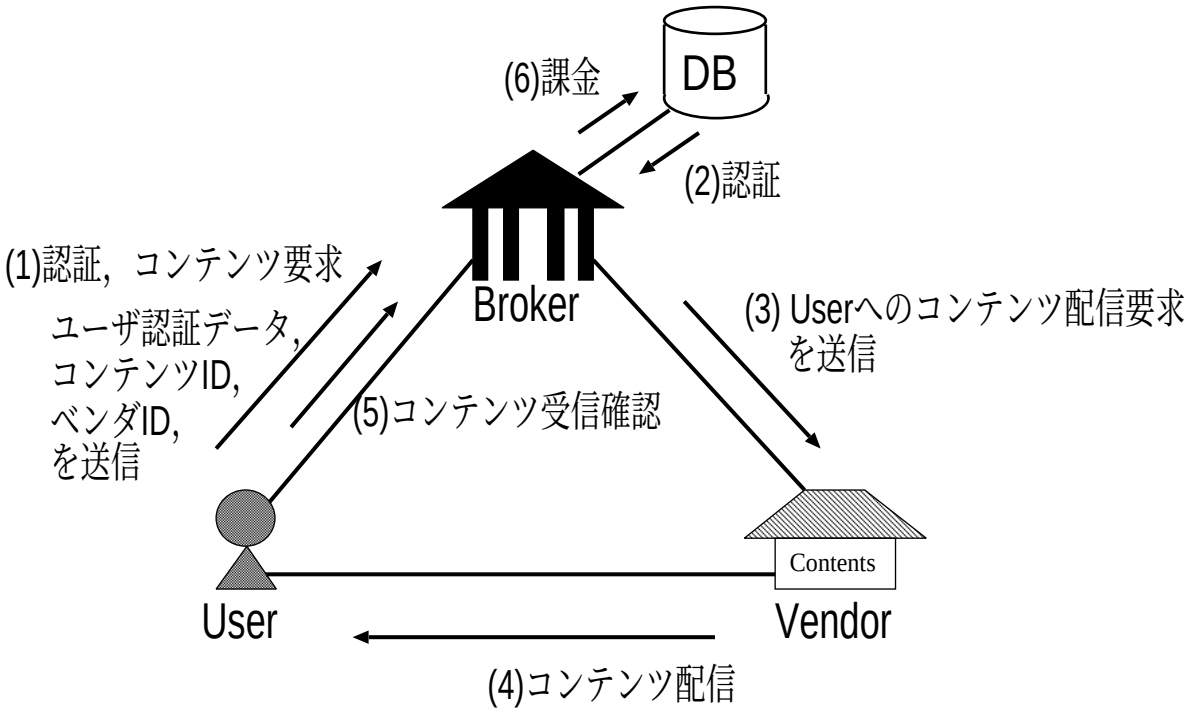
要求コンテンツ ID データ C_{ID} とコンテンツ配信要求を受信した Vendor は，User にコンテンツを送信する．

3.3.4 課金手順

コンテンツを受信した User は，コンテンツ受信確認を Broker に送信し，先程次回ユーザ認証用データ算出に用いた乱数 Nn+を登録する．

コンテンツ受信確認を受けた課金センタは，あらかじめ受信していたプロバイダ ID データ V とコンテンツ ID データ C_{ID} からコンテンツ金額 C_{CHARGE} を取得し，コンテンツ金額 C_{CHARGE} がユーザの残高 M_{USER} を上回らない時，今回残高 M_{USER} とコンテンツ金額 C_{CHARGE} の差額を残高 M_{USER} として登録し，E^2_{n+1} を次回ユーザ認証用データとして登録する．

Protocol.2



(1) User は Broker にユーザ認証データと，要求コンテンツのIDと
要求コンテンツを所持しているコンテンツベンダのIDを送信する．

(2) Userから認証要求，コンテンツ要求を受信した Broker は認証を行う．

(3) 認証が成立すれば，Vendor にUserへコンテンツを配信させるための通知をする．

(4) 通知を受けたVendor は User にコンテンツを配信する．

(5) コンテンツを一時的に受信した User は， Broker にコンテンツ受信確認を通知する．

(6) 確認を受けた Broker は，User と Vendor に対して課金を行う．

図 3.3 Prtotocol.2

3.3.5 エラー処理

認証手順において User が認証されなかった場合

Broker は User に対して「認証失敗」のメッセージを送信し，再度認証を要求する．

User はコンテンツ要求手順に戻る．

コンテンツ送信手順において，コンテンツ金額 C_{CHARGE} がユーザの残高 M_{USER} を上回る場合

Broker は User に対して「残高不足」と「認証データ更新」のメッセージを送信し，ユーザ認証データを次回ユーザ認証データに書き換える．

User は次回ユーザ認証用データ算出に用いた乱数 N_{n+1} を登録する．

第 4 章

プロトコルの考察

4.1 コンテンツ配信における信頼性

4.1.1 Protocol.1

User はユーザ認証データを Vendor に送信し，Vendor はそのユーザ認証データを Broker に送信し，認証データの正当性を検証してもらう．この際，認証方式に SAS-K を適用しているので，Vendor によるユーザ認証データへの不正は無意味である．更に，User の希望するコンテンツベンダ以外がユーザ認証データを利用することを防止するため，Broker は User とともに Vendor の認証も行う．

Vendor は Broker へ認証データを送信する際にコンテンツ URL を一緒に送信するので，Broker はコンテンツ URL を自ら送信して課金することができる．

こうしてコンテンツ URL は Broker の完璧も管理され User のもとに送信され，それに伴う課金も高い信頼性を得ることができる．

4.1.2 Protocol.2

Broker は User の認証が成立すると，User から送信されたベンダ ID データとコンテンツ ID データを基に Vendor にコンテンツ配信要求を送信する．

このプロトコルでコンテンツ配信を Broker を中継した場合としなかった場合で，コンテンツ配信の信頼性に影響を及ぼさない．なぜならば，このプロトコルでは Broker は Vendor の認証を行わないので，Vendor のベンダ ID データなどは User が送信したデータに依存す

る．そのベンダ ID データなどに誤りがあった場合，コンテンツ配信に Broker を中継しても正しいコンテンツが配信されない．つまり，Vendor が認証されないプロトコルでは，コンテンツを Vendor から直接 User に送信させ，User に受信の確認を受けて課金を行った方が，エラー処理も効率的に行える．

4.2 Broker の負荷

4.2.1 Protocol.1

Broker は User と Vendor の 2 者の認証を行う場合，Broker での処理量は必然的に増し，ユーザの増加につれ更に負荷がかかっていくと考えられる．

そこでこのプロトコルでは，User にユーザ認証の際の認証データを Vendor に送信させ，Broker とデータのやりとりを行うのを Vendor のみ，ただ一度とした．このため，Broker での処理を認証から課金，コンテンツ URL 送信まで一連に行うことができ，複雑な同期を必要としない．これは，User と Vendor の認証を別々に行う場合より遥かに Broker への負荷が軽い．

しかし，User，Vendor の認証が成立しなかった場合のエラー処理に，User とのコネクションが必要となってくる．認証不成立などのエラーは，User 数が増加すればそれに伴い増していくと考えられるので，大規模な User 数におけるコミュニケーションでの Broker への負荷は否めない．

4.2.2 Protocol.2

送受信されるデータ量が最も大きくなるのは Vendor から User に送信されるコンテンツデータで，Broker が送受信するデータ量は Protocol.1 に比べ約 5 分の 1 と少ない．また，Vendor の認証を必要せず，コンテンツ配信の中継をしないので， Broker の処理の負担は少ない．

4.3　プロトコルの適用領域

Protocol.1 は認証不成立などのエラー時における処理で，Broker の処理負担が安定しないが，コンテンツ配信に高い信頼性を求める場合に適用できると考えられる．

Protocol.2 は，コンテンツ配信の信頼性は若干落ちるが，Broker，User，Vendor 間のネットワークトラフィック負荷，Broker のトランザクション負荷を削減し，大規模な User 数に対応する安定したコミュニケーションを求める場合に適用できると考えられる．

第 5 章

おわりに

今回の研究では，プロトコルの適用領域により，1 つはコンテンツ配信の信頼性を重視し，もう 1 つは大規模な User 数に対する安定したコミュニケーションを重視し，異なる 2 つのプロトコルを設計した．

だが，Protocol.1 には，コンテンツ URL のリトライ，ブックマーク入力による，コンテンツ購入資格のない User によるコンテンツ閲覧の問題がある．Protocol.2 には，コンテンツ配信の際にコンテンツが盗聴されるといった，コンテンツそのものに対するセキュリティの問題がある．

他人への URL 譲渡や，適当な URL 入力によるコンテンツ閲覧には，コンテンツサーバーのアドレス以下を暗号化して認証成立の際に復号する鍵を配送するといった対処方法が考えられる．りトライ，ブックマーク入力によるコンテンツ閲覧には，URL に時限制をつけるなどの対処方法が考えられる．コンテンツに対するセキュリティは，コンテンツ自体に暗号をかけ，認証成立の際に復号する鍵を配送するなどして対処できる．

今後，以上のような問題をふまえて，これらのプロトコルを実装し，実証実験を行ってゆく．

謝辞

モバイルコンテンツ流通プロトコルの研究にあたり，指導教員である清水明宏助教授には，研究を進める上での助言，論文の添削などのアドバイスを頂き，研究の手助けをして頂いたことを感謝します.

清水研究室の井上富幸さんには，研究のテーマ設定において，様々な助言を頂いたことを感謝します.

同研究室の大石恭裕くんには，共通研究項目において様々な助言を頂いたことを感謝します.

その他清水研究室の皆さんには，資料の提供や，励ましの言葉を頂き，研究の手助けをして頂いたことを感謝します.

NTT-AT の渋谷さんには，研究の進め方において，様々な助言を頂いたことを感謝します.

参考文献

[1] http://www.tca.or.jp/

[2] シーメディア モバイルメディア・マガジン vol.58 第 8 巻第 4 号通巻 58 号

[3] 服部，菅野，“マイクロペイメント－デジタルコンテンツ流通のキーを握る決済手段－” 情報処理学会，Vol.39,No.1

[4] 安原，“EC の技術動向：デジタルコンテンツ作成流通技術”，情報処理学会，vol.38,No.9,Sep.1997

[5] 清水，NTT-AT，“プリペイド型課金サービス利用者の認証方法及びその初期認証データの登録方法”，特願平 11-207326

[6] 大石，林，井上，清水，“強力なパスワード認証方式の提案 (SAS-K)”，電子情報通信学会，投稿予定

[7] 堀岡，清水，“暗号化及び課金機能を有するコンテンツ提供方式の検討”，信学技報 OFS，Vol.97，No.55，pp7-12，Jun.1998

[8] 中村，“EC の技術動向：セキュリティ技術”，情報処理学会，Vol.38，No.9，Sep.1997

[9] 岸上, 阪本, “コンテンツ流通のビジネス動向”, 信学技報 IN, Vol.99, No.76, pp43-48, Nov.1999

付録 A

FEAL-8 プログラム

```
unsigned char KS[32];


makekey(key)
     unsigned char *key;
{
  unsigned char t[4],b[4],d[4];
  register i,j;


  for ( i = 0 ; i < 4 ; i++ )
    d[i] = 0;


  for ( i = 0 ; i < 8 ; i++ ) {
    for ( j = 0 ; j < 4 ; j++ ){
      t[j] = key[j+4];
      b[j] = key[j+4] ^ d[j];
      d[j] = key[j];
    }
    key[5] = key[1] ^ key[0];
    key[6] = key[2] ^ key[3];
```

```
    key[5] = sbox(key[5] + (key[6] ^ b[0]) + 1);
    key[6] = sbox(key[6] + (key[5] ^ b[1]));
    key[4] = sbox(key[0] + (key[5] ^ b[2]));
    key[7] = sbox(key[3] + (key[6] ^ b[3]) + 1);
    for ( j = 0 ; j < 4 ; j++ )
      key[j] = t[j];
    for ( j = 0 ; j < 4 ; j++ )
      KS[4*i+j] = key[j+4];
  }
}



cipher(block)
     unsigned char *block;
{
  unsigned char t[4];
  register i,j;
  int k;

  k = 16;

  for ( i = 0 ; i < 8 ; i++ )
    block[i] = block[i] ^ KS[k++];

  for ( i = 0 ; i < 4 ; i++ )
    block[i+4] = block[i+4] ^ block[i];
```

```
k = 0;

for ( i = 0 ; i < 8 ; i++ ) {
  for ( j = 0 ; j < 4 ; j++ )
    t[j] = block[j+4];
  block[5] = block[5] ^ block[4] ^ KS[k];
  block[6] = block[6] ^ block[7] ^ KS[k+1];
  block[5] = sbox(block[5] + block[6] + 1);
  block[6] = sbox(block[6] + block[5]);
  block[4] = sbox(block[4] + block[5]);
  block[7] = sbox(block[7] + block[6] + 1);
  for ( j = 0 ; j < 4 ; j++ )
    block[j+4] = block[j+4] ^ block[j];
  for ( j = 0 ; j < 4 ; j++ )
    block[j] = t[j];
  k += 2;

}

for ( i = 0 ; i < 4 ; i++ )
  t[i] = block[i+4];
for ( i = 0 ; i < 4 ; i++ )
  block[i+4] = block[i] ^ block[i+4];
for ( i = 0 ; i < 4 ; i++ )
  block[i] = t[i];
```

```
  k = 24;

  for ( i = 0 ; i < 8 ; i++ )
    block[i] = block[i] ^ KS[k++ ];

}



decipher(block)
     unsigned char *block;
{
  unsigned char t[4];
  register i,j;
  int k;

  k = 24;

  for ( i = 0 ; i < 8 ; i++ )
    block[i] = block[i] ^ KS[k++];

  for ( i = 0 ; i < 4 ; i++ )
    block[i+4] = block[i+4] ^ block[i];

  k = 14;
```

```
for ( i = 0 ; i < 8 ; i++ ) {
  for ( j = 0 ; j < 4 ; j++ )
    t[j] = block[j+4];
  block[5] = block[5] ^ block[4] ^ KS[k];
  block[6] = block[6] ^ block[7] ^ KS[k+1];
  block[5] = sbox(block[5] + block[6] + 1);
  block[6] = sbox(block[6] + block[5]);
  block[4] = sbox(block[4] + block[5]);
  block[7] = sbox(block[7] + block[6] + 1);
  for ( j = 0 ; j < 4 ; j++ )
    block[j+4] = block[j+4] ^ block[j];
  for ( j = 0 ; j < 4 ; j++ )
    block[j] = t[j];
  k -= 2;

}

for ( i = 0 ; i < 4 ; i++ )
  t[i] = block[i+4];
for ( i = 0 ; i < 4 ; i++ )
  block[i+4] = block[i] ^ block[i+4];
for ( i = 0 ; i < 4 ; i++ )
  block[i] = t[i];

k = 16;
```

```
  for ( i = 0 ; i < 8 ; i++ )
    block[i] = block[i] ^ KS[k++ ];

}

sbox(data)
     unsigned char data;
{
  return ( data << 2 | data >> 6 );

}
```