社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2004－ＣＳＥＣ－26 (53)
2004／7／21

匿名譲渡可能なオフライン型電子チケットシステム

三神 京子† 中村明日香†† 繁富 利恵††† 小川 貴英†

† 津田塾大学，東京都小平市津田町 2-1-1
†† ニイウス株式会社，東京都中央区新川 1-17-21 茅場町ファーストビル
††† 東京大学生産技術研究所，東京都目黒区駒場 4-6-1
E-mail: †{m04kmika,ogawa}@tsuda.ac.jp, ††U1694@niws.co.jp, †††sigetomi@imailab.iis.u-tokyo.ac.jp

あらまし 映画館への入場券など，様々な権利を電子化した電子チケットサービスが普及してきている．様々な情報が電子化されることで，情報の取り扱いが容易になる一方，その容易さから利用者のプライバシが脅かされる可能性がある．電子チケットサービスでも，利用者の個人情報，購入・使用履歴などの情報が電子的に管理され，サービス提供者は利用者に応じたサービスを提供しやすくなったが，このような情報が悪用されれば，利用者のプライバシ侵害につながる．そこで本論文では，チケットの入手や譲渡において，利用者の匿名性を保証しつつも適切に運用できる電子チケットシステムを提案する．本システムは，オフライン型電子現金スキームの応用の一つである Refreshable Tokens スキームの機能を拡張し，匿名でのチケットの購入・使用，さらに利用者間でのチケット譲渡を可能にする．
キーワード 電子マネー，プライバシ保護，電子チケット

Transferable Off-line Anonymous Electronic Tickets System

Kyoko MIKAMI†, Asuka NAKAMURA††, Rie SHIGETOMI†††, and Takahide OGAWA†

† Tsuda College 2-1-1 Tsuda-matchi, Kodaira-shi, Tokyo
†† NIWS Co.,Ltd Kayabacho-First Bldg., 1-17-21 Shinkawa, Chuo-ku, Tokyo
††† Institure of Industrial Science, the University of Tokyo 4-6-1 Komaba, Meguro-ku, Tokyo
E-mail: †{m04kmika,ogawa}@tsuda.ac.jp, ††U1694@niws.co.jp, †††sigetomi@imailab.iis.u-tokyo.ac.jp

Abstract Electronic ticket services, such as tickets for movies, are begin popular. It will be easy to deal with the information when many kinds of information are digitized. On the other hand, there is a possibility of invading the user's privacy with that easiness. In the electronic ticket services, by storing personal information, purchase records, and past records of users' usage, the service providers could provide services suitable for the user's needs. However, if these information were misused, it will lead to the privacy infringement. To resolve these problems, this paper proposed an electronic tickets system that will not allow the service provider to get the information who bought what. This system extends Refreshable Tokens Scheme applying off-line electronic cash scheme enables us not only to buy and use tickets but also to transfer them to other users anonymously.
Key words Electronic Cash, Privacy Protection, Electronic Ticket

1. はじめに

今日，様々な場面で情報の電子化がすすむにつれて，多くのサービスでも電子情報が利用されるようになった．中でも映画，コンサートなどの入場券を扱う電子チケットサービスは電子化が進んだ一つの普及例といえる．このサービスでは，電子化されたチケットをインターネット上で流通させることにより，チケット受け取りの手間や印刷コストなどを低減することを可能にした．また，電子情報は複製，検索，統合が容易な為，個人情報と利用者の過去の購入・使用履歴を利用することによって，趣味や興味などに応じた，よりきめ細やかなサービスを行いやすくなった．たとえば，ある利用者がサッカーの観戦チケットを買ったとし，後日同じチームのサッカー観戦チケットを買ったとする．この情報から，サービス提供者はその利用者はそのチームが好きに違いないと推測することができ，その利用者にそのチームのチケットを紹介したり，優先的に販売したりするサービスを提供することができる．しかし，こういった情報を一つの機関が一括管理するようになると，その情報が漏洩した場合の被害は計り知れない．

ネットワークを介した電子現金での支払いが可能であると仮

	使用時に現金(チケット)発行者へのアクセスの有無	支払い・譲渡時に現金(チケット)発行者へのアクセスの有無	問題点	
オンライン型電子現金[3]	する		電子現金使用時に2重使用検証のため，検証機関へのアクセスが必要となる	
オフライン型電子現金[3]	しない	する	受け取った電子現金は一度自分の口座へ入金し，必要になった時に引き出す．	電子チケットでこの方式を採用すると，サービス提供者はどの口座にどのチケットが入ったのかという情報を得ることができるため，誰がチケットを手に入れたのかという情報を知ることができる．
提案方式	しない	する	チケットの譲渡は，口座を介すことなくできる．	この方式では，サービス提供者は，誰がチケットを入手したのか知ることができない．
オフライン型電子現金(耐タンパデバイス有)[1][2]	しない	しない	耐タンパデバイスを必要とする	
オフライン型電子現金(耐タンパデバイス無)[4]	しない	しない	耐タンパデバイスは必要としないが，譲渡が起こるたびに電子現金サイズが大きくなる	

図 1　電子現金と電子チケット

定したならば，サービス提供者はどの利用者がチケットを購入したのかという情報は必ずしも必要ではない．なぜなら，サービス提供者は，チケットをもつその人がサービスを受ける権利があるということさえ証明されればよくとくに誰であるということを特定する必要はないからである．必要のない情報は最初から手に入れない・与えないということは大切である．そこで私たちは，サービス提供者が誰がチケットを入手・譲渡したのかという情報を持たずに，適切に運用できる電子チケットシステムを提案する．

1.1　電子現金技術と電子チケット

電子現金では複製や偽造といった不正を防止すると同時に，現金使用時における利用者の匿名性を保証しなくてはならない．電子チケットでも，複製や偽造を防止することは当然である．しかし，電子チケットは，その実現方法として電子現金方式を利用したとしても，誰にどのチケットを発行したのかという情報さえあれば，誰がチケットを使用しているのかを，推測もしくは特定できる可能性がある．そこでまず，従来の電子現金スキームを電子チケットに適用した場合の問題点などを考察してみる．

電子現金にはオンライン型電子現金とオフライン型電子現金がある．オンライン型電子現金 [3] では，一度使用された現金は電子現金発行機関に蓄積されており，利用者が現金を使う際にこの蓄積されたデータを使って検証することにより，二重使用を防いでいる．しかし，この方式では，電子現金の使用のたびに電子現金発行機関へのアクセスを必要とするので，もし発行機関に何らかの理由で接続できないときには，電子現金を使用することができない．

一方，オフライン型電子現金には耐タンパデバイスを必要とする方式 [1], [2] としない方式 [3] がある．耐タンパデバイスには電子現金の複製ができないという仮定があるので，このデバイスを使用する方式では二重使用は考慮にいれない．耐タンパデバイスを必要としない方式は，オンライン型電子現金と同様に，発行機関に蓄積されたデータに基づいて二重使用の検証を行う．オンライン型電子現金との相違点はオフライン型電子現金では，現金データの中に利用者を識別するための ID を埋め込んであることである．この ID を利用して，後から不正者追跡することが可能なので，電子現金使用時に二重使用の検証を行う必要はなく，常に現金発行機関が接続できる環境にある必要はない．ただし，現金使用時における利用者の匿名性も保証されているため，この利用者 ID は通常使用ではサービス提供者には分からない．

電子チケットの場合も，使用のたびに現金発行機関へアクセスする必要のあるオンライン型電子現金と同様の方式を用いては負担が大きい．また，耐タンパデバイスを必要とする方式は，利用者にデバイスを所持させるコストや手間がかかり，またネットワークを介してサービスが提供されるような状況が想定される電子チケットサービスには向かない．耐タンパデバイスを必要としないオフライン型電子現金方式でも現金の譲渡の現金サイズが大きくなるという欠点もある [4]．(表 1 参照)

本方式の特徴は大きく次の 3 点である．1 つ目はチケットの購入・使用は匿名でできることである．2 つ目はチケット発行者への匿名なアクセスを通して，チケットの譲渡が可能であることである．3 つめは譲渡を通してチケットサイズが一定であることである．

1.2　提 案 方 式

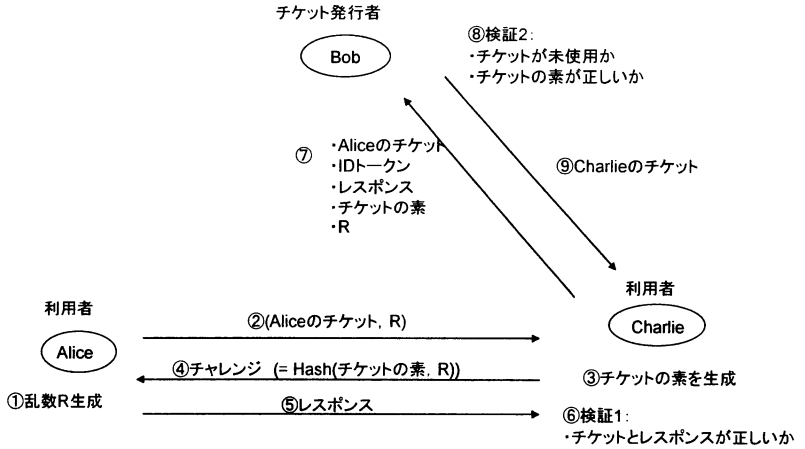
本方式では，オフライン型電子現金スキームの応用の一つである，繁富ら [8]～[10] が提案した Refreshable Tokens スキームを用いる．Refreshable Tokens スキームにおけるトークン（権利証）の特性は以下のとおりである．

（1）　偽造不可能

（2）　二重使用が行われた場合のみ不正使用者を追跡可能

（3）　二重使用が起こらない限りトークン使用時における利用者の匿名性を保証

（4）　トークンに ID を埋め込む際に，自分の ID そのものを提示するのではなく，すでにサービス提供者から発行されているトークンから新しいトークンを得ることができる．



チケットの素: CharlieのID情報を含んだチケットを作るにあたって必要な情報
IDトークン: サービス提供者がブラインド署名をしたID情報を含むトークン.
サービス提供者には誰のIDトークンであるのか分からない

図 2 Alice から Charlie への譲渡

(1) - (3) は耐タンパデバイスを用いないオフライン型電子現金でも実現されているが, Refreshable Tokens スキームでは, あらたに (4) の機能がある. オフライン型電子現金スキームでは ID を提示することによって, トークンに埋め込む ID の正当性を検証していたのに対し, Refreshable Tokens スキームでは, ID ではなく発行済みトークンを使って同様の検証が可能である. 我々はこのスキームを利用して, 以下のような手順をとることにより匿名でのチケット購入・使用・譲渡を可能にした.

まず, あらかじめサービス提供者は利用者に対して, 利用者 ID を証明するためのトークンを発行する (ID トークン). ID トークンも電子現金と同様に利用者 ID を含む権利証であり, トークンはブラインド署名によってサービス提供者が署名しているので, 誰がこのトークンを実際に使用しているのかサービス提供者には把握することができない.

チケット購入の際は ID トークンをサービス提供者に提示して行う. 従来方式ではここで ID そのものをサービス提供者に示す必要があったのだが, 支払いが電子現金といった方法で済ますことが可能であると仮定するならば, ID トークンを利用することによって, チケットを匿名で発行することが可能になる.

また, 電子現金の場合は利用者同士での譲渡が起こっても, 受け取った現金を一度銀行口座へ入金して, 口座から引き出す際に ID を示すという方式でも問題はなかったのだが, 電子チケットの場合, 誰がどのようなチケットを得ようとしているのかをサービス提供者に知られないようにするためには, ID を見せる方式では不可能である. そこで, 譲渡の際も ID トークンを利用してチケットに埋め込む ID を変更する.

匿名でチケットの発行が可能なことによって, 電子現金と同様に匿名でチケットの使用が可能になる. また, チケットの利用者同士での譲渡の場合も同じである. ここでは利用者 Alice から利用者 Charlie への譲渡を例にとって説明する. Bob はチケット発行者であり, 譲渡の際には Charlie は Bob へのアクセスを必要とする. Alice はすでにチケットを購入しているとする. もちろん匿名で購入している. これを Charlie に譲渡するのだが, Charlie も購入と同様に ID トークンを利用して自分の ID を含むチケットを得る. (図 2 参照)

1.3 論文の構成

以下, 2 章では準備として, Rrefreshable Tokens スキームについて述べる. 3 章では提案する電子チケットシステムの概要, 定義やプロトコルについて述べる. 4 章ではまとめを述べる.

2. Refreshable Tokens スキーム

この章では準備としてオフライン型電子現金スキームを応用した Refreshable Tokens スキームについて述べる.

Rereshable Tokens スキーム [8]〜[10] は, オフライン型電子現金スキームを応用した匿名認証スキームである. このスキームでは, サービス提供者が発行するトークン (権利証) を用いて利用者認証を行う. トークンは, 電子現金と同じように, 使用時における利用者の匿名性を保証している. Refreshable Tokens スキームでは, 匿名性を保証した状態で権利の更新が可能である, これは, 従来のオフライン型電子現金スキームでは, 現金を発行する際に, 利用者を識別するための ID を必要したのだが, Refreshable Tokens スキームでは, ID の代わりにすでに発行してあるトークンを利用することで, 新しいトークンを得ることが可能になったためである.

2.1 ブラインド署名

電子現金やRefreshable Tokensスキームでは，利用者が電子現金やトークンを使用する際の匿名性を保証している．これを可能にした技術の一つがブラインド署名[3]である．ブラインド署名は，通常の署名と同様に，署名の正当性を署名者の公開鍵によって検証することができる．通常の署名法とは異なる点は，ブラインド署名では，署名プロトコルの際に署名者が利用者から得られる情報と，最終的に利用者が得た署名を対応づけることができないことである．電子現金とは銀行のデジタル署名がついたデータであり，ブラインド署名を用いて銀行が署名を行えば，銀行が現金発行の際に利用者から得た情報と，最終的に利用者が手に入れた電子現金とを対応づけることができないので，現金使用時における利用者の匿名性を保証している．

2.2 定　　義

Refreshable Tokensスキームにおける参加者とその役割は以下のとおりである．

利用者：トークンをサービス提供者から発行してもらうことによってサービスを受ける権利を所有することができる．実際に，利用者がサービスを受けたいときに，サービス提供者に対してトークンを「使用 (Present)」することによって権利を持つことを証明する．

サービス提供者：トークンを発行し，利用者がトークンを「使用 (Present)」する際には検証を行う．権利の発行をする機関と検証する機関は同一である．

Refreshable Tokensスキームは以下の4つの関数によって定義される．tは利用者がサービス提供者から発行してもらうトークンを表す．idはサービス提供者が利用者を一意に識別するため情報であり，利用者ごとに異なったidが割り当てられる．また，サービス提供者の公開鍵をPK，秘密鍵をSKとする．

- *RT_Generation*：入力 (id, PK, SK)，出力 (t, w)

*RT_Generation*では，サービス提供者が利用者ごとに割り当てたidを用いて，利用者に擬似トークンを発行する．しかし，この擬似トークンは「すでにこの利用者にはidを割り当てた」ということを証明するだけのトークンであり，擬似トークンは使用時における利用者の匿名性を保証していない．そのため，擬似トークンを得た利用者は匿名性のあるトークンtを得るために，まず*RT_Refresh*をする．利用者はtとwを得る．wはidを計算するための必要な情報であり，wが分かればidを知ることができる．

- *RT_Refresh*：入力 (t, w, PK, SK)，出力 (t', w')

*RT_Refresh*では，利用者が所持しているトークンを新しいトークンに更新する．既に発行されているトークンtとそれに対応したwを*RT_Refresh*の入力とすれば，利用者は新しいトークンt'とそれに対応したw'を得ることができる．wはtごとに異なるが，同じ利用者に発行した2つのトークンをt, t'とした場合，これに対応するw, w'から計算できるidは同一である．

- *RT_Present*：入力 (t, w)，出力 *valid* または *invalid*

*RT_Present*では利用者がサービス提供者に対してサービスを受ける権利をもつことを証明する．権利を持つことを確かめることができた場合は*valid*，そうでなければ*invalid*である．

- *RT_Trace*：入力 t，出力 *id*

ある利用者が同じwを含むtを2回*RT_Present*をした場合に限り，*RT_Present*のときに利用者から得られる情報から，サービス提供者はその利用者に割り当てたidを得ることができる．

2.2.1 Refreshable Tokensスキームのセキュリティ要件

Refreshable Tokensスキームは以下の要件が満たされている．

- Unlinkability：*RT_Refresh*によって得たトークン2つt, t'あった場合，t, t'が同じ利用者が所持するトークンであるのかは知ることができない．
- Double-Use Traceability：もし，ある利用者が*RT_Present*の引数として同じtが2回使用 (二重使用) されたならば，*RT_Trace*によりその利用者に割り当てたidを計算することが可能である．
- Unforgeability：*RT_Generation*，*RT_Refresh*によって得られるトークンは偽造が不可能である．
- Refreshability：*RT_Refresh*では，サービス提供者は利用者に割り当てたidを得ることなく新しいトークンを発行することができる．

2.3 記　　法

プロトコルの説明をするにあたり必要な記号のを定義する．

- w：wはidを計算するために必要な情報であり，wを知ることができればidを求めることができる．
- m：mはwをもとに生成される．しかし，mからwを多項式時間アルゴリズムで求めることは困難である．

関数は以下のように定義する．

- *Blind*：入力 (m, r)，出力 $\mathcal{B}_m$

mをブラインドする．rはランダムに選んだ数値．rを知らなければ，$\mathcal{B}_m$からmを計算すること (*Unblind*) はできない．

- *Sign*：入力 $(\mathcal{B}_m, SK, PK)$，出力 $\mathcal{S}_{\mathcal{B}_m}$

サービス提供者の公開鍵PKと秘密鍵SKを用いて，$\mathcal{B}_m$にブラインド署名を行う．

- *Unblind*：入力 $(\mathcal{S}_{\mathcal{B}_m}, r)$，出力 $\mathcal{S}_m$ または *invalid*

mをブラインドする際に用いた乱数rが分かれば$\mathcal{S}_m$を得ることができる．できなかった場合は*invalid*.

- *R_Check*：入力 $(t, \mathcal{B}_{m'})$，出力 *valid* または *invalid*

$\mathcal{B}_{m'}$とtが同じIDに基づいて生成されているかの検証を，サービス提供者は直接m'の情報を得ることなく，利用者の助けをかりて検証する．同じであることを確かめることができれば*valid*，そうでなければ*invalid*となる．

- *Verify*：入力 (PK, t)，出力 *valid* または *invalid*

tにされている署名が本当にサービス提供者の秘密鍵によってされたものか公開鍵PKを用いて検証する．署名が正しいことが検証されれば*valid*，そうでなければ*invalid*となる．

- *Responce*：入力 (α, w)，出力 β

ランダムに選ばれたチャレンジ α から *RT_Generation* や *RT_Refresh* の出力である w を使ってレスポンスを計算する．β から w は求めることはできない．ただし，同じ t が *RT_Present* された場合 *RT_Trace* の入力条件がととのえば id を求めることが可能である．

- *P_Check*：入力 (t, β, α)，出力 *valid* または *invalid*

t を生成するために使われた w と *Responce* 関数によって得られる β を生成するために使われた w が同一であるかの検証をする．同じであることが検証されれば *valid*，そうでなければ *invalid*.

2.4 プロトコル

ここではRefreshable Tokensスキームで定義された4つの関数 *RT_Generation*, *RT_Refresh*, *RT_Present*, *RT_Trace* のプロトコルを述べる．なお以下の説明ではAlice(利用者)，Bob(サービス提供者)とする．Bobが Aliceに対して割り当てた id を id_a とする．

RT_Generation：Bobは Aliceに割り当てた id を使って，匿名性のない擬似トークン t_{p_a} を発行する．Aliceが擬似トークンを *RT_Refresh* することで，サービスを受けるために必要な匿名性のあるトークンを得ることができる．

（1） Bobは利用者 Aliceに割り当てた id_a から w_{p_a} を生成し，さらに w_{p_a} から m_{p_a} を生成．

（2） さらに Bobは $\mathcal{S}_{m_{p_a}} \leftarrow Sign(m_{p_a}, SK, PK)$ と署名をする．擬似トークン $t_{p_a} = (m_{p_a}, \mathcal{S}_{m_{p_a}})$ と w_{p_a} を Aliceに送信．

（3） Aliceは t_{p_a} を *RT_Refresh* することによって t_a を得ることができる．

RT_Refresh：すでに利用者が所持しているトークンから新しいトークンを発行する．ここでは，Aliceはすでにトークン $t_a = (m_a, \mathcal{S}_{m_a})$ が発行されていることとする．

（1） Aliceは t_a に対応した w_a から w'_a を計算．さらに w'_a から m'_a を生成．

（2） Aliceはランダムに r' を選び，$\mathcal{B}_{m'_a} \leftarrow Blind(m'_a, r')$ とする．$\mathcal{B}_{m'_a}$ と t_a を Bobに送信．

（3） Bobは $R\_Check(t_a, \mathcal{B}_{m'_a})$ が *valid* ならば，$\mathcal{S}_{\mathcal{B}_{m'_a}} \leftarrow Sign(\mathcal{B}_{m'_a})$ とすることで，$\mathcal{B}_{m'_a}$ に対して署名をする．$\mathcal{S}_{\mathcal{B}_{m'_a}}$ を Aliceに送信．

（4） Aliceは $Verify(\mathcal{S}_{\mathcal{B}_{m'_a}}, PK)$ が *valid* ならば，$\mathcal{S}_{m'_a} \leftarrow Unblind(\mathcal{S}_{\mathcal{B}_{m'_a}}, r')$ とすることでブラインドをはずす．Aliceは $t'_a \leftarrow (m'_a, \mathcal{S}_{m'_a})$ を得る．

RT_Present：トークンを所持する利用者がそのトークンに応じたサービスを受ける権利をもつものか，証明・検証する．Aliceはすでに $t_a = (m_a, \mathcal{S}_{m_a})$ をサービス提供者から発行されているものとする．

（1） Aliceは Bobにトークン t_a を送信．

（2） Bobは $Verify(t_a, PK)$ が *valid* である場合，ランダムに選んだチャレンジ α を Aliceに送信．

（3） Aliceは $\beta \leftarrow Responce(\alpha, w_a)$ を Bobに送信．

（4） Bobは $P\_Check(t_a, \alpha, \beta)$ が *valid* ならば，t_a を受理する．

RT_Trace：利用者によってトークンが二重使用されたときのみ，サービス提供者はその利用者に割り当てた id を知ることができる．サービス提供者は *RT_Present* されたトークンをすべて蓄積し，この蓄積データに基づき二重使用の検出をする．さらに，誰が二重使用したのかを特定するために，*Responce* 関数の入力データと出力データを使用する．2度の *RT_Present* によって得られる，*Responce* 関数の入力データ α, α' と，それに対応する出力データ β, β' より，利用者に割り当てた id の計算ができる．これによって2重使用した利用者を特定することができる．

3. 電子チケットシステム

この章では提案する電子チケットシステムについて述べる．なお，システムの詳細については第1.章を参照のこと．

3.1 定　　義

このシステムにおける参加者とその役割は以下のとおりである．

利用者：チケット発行者からチケットを発行してもらうことによって，チケットに応じたサービスを受ける権利を持つことができる．実際に，利用者がサービスを受けたいときには，サービス提供者に対してトークンを「使用 (Present)」することによって権利を持つことを証明する．また，利用者同士でのチケットの譲渡も可能である．

チケット発行者：サービス提供者の代理として，利用者にチケットを発行する機関．

サービス提供者：チケットが使用されたときに検証を行う．チケットに応じたサービスを提供する．

電子チケットは以下のつのプロトコルで構成されることを定義する．なお以下の説明ではAlice, Charlie(利用者)，Bob(チケット発行者)，Dave(サービス提供者)である．また，チケットの種類を表すためのコード *code* とした場合，PK_{code} をチケットコード *code* に対応したチケット発行者の公開鍵であり，SK_{code} を秘密鍵とする．なお Refreshable Tokens スキームと同様に定義される記号については省略する．

- *DT_Registraion*：

入力 (id, PK_{id}, SK_{id})，出力 (t_{id}, w)

DT_Registraion では，利用者に対して ID トークン t_{id} の発行を行う．t_{id} はその利用者にはチケットを入手する権利があることを示すためのトークンである．

- *DT_Issue*：

入力 $(code, t_{id}, PK_{code}, SK_{code})$，出力 (t'_{code}, w')

DT_Issue では，チケット発行者が利用者にコードに応じたチケットを発行する．チケットコード *code* に対応した署名がさ

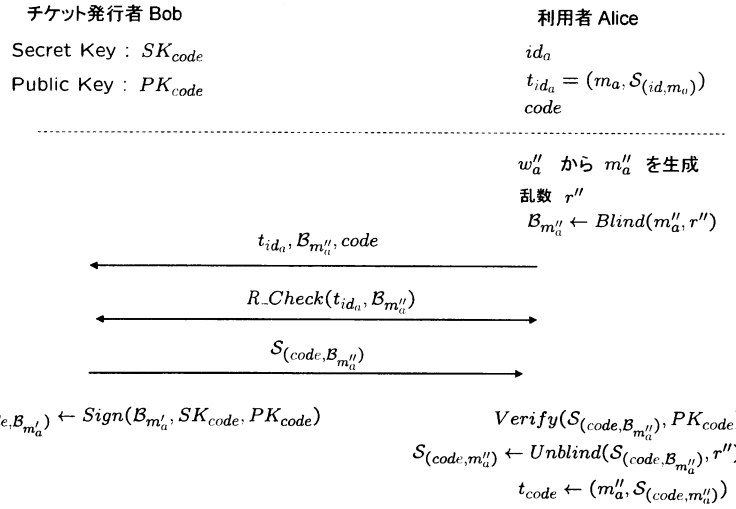


図 3 Ticket Generation

れたチケットを利用者は得ることができる.

- *DT_Present* : 入力 (t_{code}, w), 出力 *valid* または *invalid*

DT_Present では利用者がサービス提供者に対してチケットコード *code* に応じたサービスを受ける権利をもつことを証明する. それを確かめることができた場合の出力は 1, そうでなければ 0 である.

- *DT_Transfer* :
  入力 $(t_{code_a}, id_c, t_{id_c}, PK_{code}, SK_{code})$,
  出力 $(w_c, invalid$ または $t_{code_c})$

たとえば, Alice が所有するチケット t_{code_a} を, Charlie が使用できるチケット t_{code_c} に変更をする. 変更に失敗した場合の出力は 0 である.

- *DT_Trace* : 入力 t_{code}, 出力 *id*

ある利用者が同じ t_{code} や t_{id} を 2 回 *DT_Present* した場合, サービス提供者はその利用者に割り当てた *id* を得ることができる. また, 一度でも *DT_Present* したチケットを *DT_Transfer* をした場合も *id* を割り出すことが可能である. 逆も同様である.

3.2 電子チケットシステムのセキュリティ要件

提案する電子チケットシステムは以下の要件を満たさなくてはならない.

- Unforgeability : *DT_Registraion*, *DT_Issue*, *DT_Transfer* によって得られるチケットや ID トークンは偽造不可能である.
- Transferability : *DT_Transfer* によって t_{code} を使用できる利用者を変更することができる. このことを譲渡と呼ぶ. いったんチケットを譲渡したら, 同じように譲渡されない限り, もとの持ち主はそのチケットを使用する権利を失う. 新しい所有者となる者は自分の ID トークンと *RT_Refresh* をしないかぎり, チケットを使用することができない. 新しい所有者以外にはチケットを使用することができない.
- Unlinkability : *DT_Registratin* や *DT_Issue*, *DT_Refresh* で得られる 2 つのトークンを t, t' とした場合, t, t' が同じ利用者が所有するトークンであるかを知ることができない.
- Traceability : もし *RT_Present*, *DT_Present* の引数として同じ t_{code} が 2 回使用されたとき, また, *DT_Transfer* において, 一度でも *RT_Present* されたチケットを入力引数とした, あるいは, 一度でも *RT_Present* したチケットを *DT_Transfer* をした, のいずれかの場合に当てはまるときのみ, サービス提供者は利用者の *id* を計算することができる.

3.3 プロトコル

ここでは, 電子チケットシステムで定義された 5 つの関数 *DT_Registraion*, *DT_Issue*, *DT_Present*, *DT_Transfer*, *DT_Trace* のプロトコルを述べる. 以下の説明では, Alice, Charlie(利用者), Bob (チケット発行者), Dave(サービス提供者) とする. チケット発行者が, 利用者 Alice,Charlie に対して割り当てた *id* を id_a, id_c とする.

DT_Registration : チケット発行者は利用者を識別するために割り当てた *id* をもとに ID トークンの発行を行う. ID トークンがなくては, チケットの購入や譲渡はできないため, 利用者は必ずチケット発行者から ID トークンの発行をしてもらう. *RT_Generation* に対して, 入力 $(id, PK_{code}, SK_{code})$ によって出力 $t_{id_a} = (m_a, \mathcal{S}_{(id,m_a)})$ を ID トークンとする.

DT_Issue : 利用者はチケット発行者からチケットの発行をしてもらう. ここでは, Alice は ID トークン $t_{id_a} = (m_a, \mathcal{S}_{(id,m_a)})$ を使って *RT_Refresh* することによりチケットの得ることができる. (図 3 参照)

(1) Alice は t_{id_a} に対応した w_a から w''_a を計算. さらに w''_a から m''_a を生成.

(2) Alice はランダムに r'' を選び, $\mathcal{B}_{m''_a} \leftarrow Blind(m''_a, r'')$ とすることで, m''_a をブラインドする. $\mathcal{B}_{m''_a}$ と t_{id_a} と希望する

チケットコード $code$ を Bob に送信.

（3） Bob は $R\_Check(t_{id_a}, \mathcal{B}_{m''_a})$ が $valid$ ならば，$\mathcal{S}_{(code_\epsilon, \mathcal{B}_{m''_a})} \leftarrow Sign(\mathcal{B}_{m''_a}, SK_{code})$ とすることで，$code$ に応じた署名をする．$\mathcal{S}_{(code_\epsilon, \mathcal{B}_{m''_a})}$ を Alice に送信する．

（4） Alice は $Verify(\mathcal{S}_{(code_\epsilon, \mathcal{B}_{m''_a})}, PK_{code_\epsilon}))$ により，署名が正しくされていることを確かめることができたら，$\mathcal{S}_{(code_\epsilon, m''_a)} \leftarrow Unblind(\mathcal{S}_{(code_\epsilon, \mathcal{B}_{m''_a})}, r'')$ とすることでブラインドをはずす．$t''_{code_{e_a}} \leftarrow (m''_a, \mathcal{S}_{(code_\epsilon, m''_a)})$ がチケットコード $code$ のチケットとなる．

（5） 最後に Alice は t_{id_a} を $RT\_Refresh$ する．

DT_Present：チケットを所持する利用者 Alice がそのトークンに応じたサービスを受ける権利をもつものか，サービス提供者 Dave に対して証明する．Alice はすでに $t''_{code_{e_a}} = (m''_a, \mathcal{S}_{(code_\epsilon, m''_a)})$ をサービス提供者から発行されているものとする．

（1） Alice は $t''_{code_{e_a}}$ を Dave に送信する．

（2） Dave は $Verify(t''_a, PK_{code})$ により，，Alice にランダムに選んだチャレンジ α を送信．

（3） Alice は $\beta \leftarrow Responce(w''_a, \alpha)$ を計算し，Bob に送信．

（4） Bob は $P\_Check(t''_{code_{e_a}}, \alpha, \beta)$ が $valid$ ならば，トークンに応じたサービスを提供する．

DT_Transfer：利用者 Alice ら利用者 Charlie へチケット譲渡をする．Alice はすでにチケット $t''_{code_{e_a}} = (m''_a, \mathcal{S}_{(code_\epsilon, m''_a)})$ を所持しており，まだ使用していないものとする．Charlie は ID トークン $t_{id_c} = (m_c, \mathcal{S}_{(id, m_c)})$ を所持しているものとする．また，利用者，チケット発行者共通の一方向性ハッシュ関数を $\mathcal{H}$ と定義する．$\mathcal{H}$ では入力はすべて連結し，ハッシュする．

（1） Alice は譲渡したいチケット $t''_{code_{e_a}}$ とランダムに選んだ数値 R を Charlie に送信する．

（2） Charlie は t_{id_c} に対応した w_c から w'_c を計算し，さらに w'_c から m'_c を生成する．

（3） Charlie はランダムな数値 s を選び，$\mathcal{B}_{m'_c} \leftarrow Blind(m'_c, s)$ と計算．以下のように乱数 R とハッシュする．

$$\alpha_c \leftarrow \mathcal{H}(\mathcal{B}_{m'_c}, R)$$

α を Alice に送信．

（4） Alice は $\beta_a \leftarrow Responce(w''_a, \alpha_c)$ を計算し，Charlie に送信．

（5） Charlie は $P\_Check(t''_{code_{e_a}}, \alpha_c, \beta_a)$ が $valid$ ならば，以下を Bob にすべて送信する．

$$(t_{id_c}, t''_{code_{e_a}}, R, \beta_a, \mathcal{B}_{m'_c})$$

（6） Bob は $R\_Check(t_{id_c}, \mathcal{B}_{m'_c})$ が $valid$ ならば，次に Bob は以下のような検証を行う．

$$P\_Check(t''_{code_{e_a}}, \mathcal{H}(\mathcal{B}_{m'_c}, R), \beta_a)$$

検証した結果が 1 であれば，$\mathcal{S}_{(code_\epsilon, \mathcal{B}_{m'_c})} \leftarrow Sign(\mathcal{B}_{m'_c}, SK_{code_\epsilon})$ を Alice に送信する．

（7） Charlie は $Verify(\mathcal{S}_{(code_\epsilon, \mathcal{B}_{m'_c})}, PK_{code_\epsilon})$ によって署名が正しいことを検証したら，$\mathcal{S}_{(code_\epsilon, m'_c)} \leftarrow Unblind(\mathcal{S}_{(code_\epsilon, \mathcal{B}_{m'_a})}, s)$ とすることによってブラインドをはずし，$t_{code'_c} \leftarrow (m'_c, \mathcal{S}_{(code_\epsilon, m'_c)})$ を得る．

（8） 最後に Charlie は $RT\_Refresh(t_{id_c})$ をする．

DT_Trace：利用者によってチケットが二重使用されたとき，サービス提供者はその利用者に割り当てた id を知ることができる．$RT\_Present$ と同様，チケット発行者は $DT\_Present$ されたチケットをすべて蓄積し，この蓄積データをもとに二重使用の検出をする．さらに，誰が二重使用したのかを特定するために，$Responce$ 関数の入力データと出力データを使用する．2 度の $DT\_Present$ によって得られる，$Responce$ 関数の入力データ α, α' と，それに対応する出力データ β, β' より，利用者に割り当てた id の計算ができる．これによって 2 重使用した利用者を特定することができる．

また，本システムでは，$DT\_Transfer$ の際にチケット発行者が得ることのできるもとのチケット所持者のチケットデータも必ず蓄積する．このことにより，すでに使用されたチケットを譲渡した場合や譲渡したチケットを使用した場合の不正者の追跡を可能にしている．

3.4 検　　証

ここでは，Transferability と Traceability の検証を行う．Unlinkability，Unforgeability については [8]～[10] 参照のこと．考えられる攻撃としては以下のケースがあげられる．

Case1：攻撃者が他の利用者にチケット t_{code} を譲渡したにも関わらず，攻撃者が t_{code} を $DT\_Present$ する．

Case2：攻撃者が一度 $DT\_Present$ した t_{code} を他の利用者に譲渡する．

Case3：ある利用者が所持する t_{code} を攻撃者が受け取った場合，$RT\_Refresh$ をすることなく $DT\_Present$ する．

Case1，Case2 のどちらの場合も，$DT\_Transfer$ と $DT\_Present$ の過程において，Bob は $t_{code_{e_a}}$ とチャレンジ α，α に対する $Responce$ 関数の出力 β を得ることができる．そのため，Case1，Case2 のような不正が行われたとしても，DT_Trace により攻撃者の id を計算することができる．Case3 は，$DT\_Present$ する際に必要な w を攻撃者は知ることができないため，他の利用者から得た t_{code} を $RT\_Present$ するのは不可能である．

4. おわりに

提案方式によってチケットの購入・使用・利用者同士での譲渡を匿名で行うことができた．今回の提案では，Refreshable Tokens スキームにおけるトークン（権利証）が持つデータの中に，時間指定や座席番号といったチケット固有の情報を含ませることが可能であると考えたが，その証明はまだ終わっていない．そのため，この証明をするということが今後の課題となる．また，実際に電子チケットシステムとして運用していくためには，チケット固有情報をどのように持てば，よりチケット

として扱いやすいくなるかなどについても考えていかなくてはならない.

文　献

[1] S.Brands. Untraceable off-line cash in wallet with observers. In Advances in Cryptology–CRYPTO '93, volume 911, pages 302-318. Springer-Verlag, 1994.

[2] S.Brands. Off-line Cash Transfer by Smart Cards

[3] D.Chaum, A.Fiat, and M.Naor. Untraceable Elecronic Cash. Proc. of CRYPTO '88, pp.319-327, 1988

[4] D.Chaum and Torben Pryds Pedersen. Transferred cash grows in size. In Advances in Cryptology–Proceedings of EUROCRYPT '92, volume 658 of LNCS, pages 390-407. Springer-Verlag, May 1992.

[5] M.Terada, H.Kuno, M.Hanadate, and K.Fujimura, "Copy Prevention Scheme for Rights Trading Infrastructure", 4th Smart Card Reserch and Advanced Application Conference(CARDIS 2000), September 2000.

[6] 花舘蔵之, 寺田雅之, 千綿伸之, 水野康尚: スマートカードを利用した電子商取引のための原本性保証システムの開発, コンピュータセキュリティシンポジウム'01 予稿集 (2000). October 2000

[7] K. Fujimura, H. Kuno, M. Terada, K. Matsuyama, Y. Mizuno, and J. Sekine, "Digital-Ticket-Controlled Digital Ticket Circulation", 8th USENIX Security Symposium, August 1999, pp. 229-238

[8] R. Shigetomi and A. Otsuka and T. Ogawa and H. Imai. Refreshable Tokens. 情報理論とその応用シンポジウム 2002. 伊香保

[9] R. Shigetomi and A. Otsuka and T. Ogawa and H. Imai.Refreshable Tokens and its application to Anonymous Loan.SCIS 2003. 南紀白浜

[10] R. Shigetomi, A. Otsuka, and H. Imai. Anonymous authentication scheme for xml security standard wh refreshable tokens. In ACM Workshop on XML Security, October, 2003.