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

2003－ＣＳＥＣ－22　(34)
2003／7／18

P2Pクレデンシャルフレームワークの提案

川口　信隆†　小畑　直裕†　宮地　玲奈†　上田真太郎†　重野　寛†
岡田　謙一†

† 慶應義塾大学理工学部　〒223-8522 神奈川県横浜市港北区日吉 3-14-1
E-mail: †{kawaguti,obata,miyaji,ueda,shigeno,okada}@mos.ics.keio.ac.jp

あらまし　クレデンシャルは電子マネー，電子チケットに代表される電子的な価値である．本論文では様々な形態のクレデンシャルをオンライン・オフラインの両方で使用するためのP2Pフレームワークを提案する．また，オンライン上でクレデンシャルの委譲を行うための共同体フレームワークについて考察する．
キーワード　クレデンシャル，ピア・ツー・ピア，スマートカード，委譲

A Proposal of P2P Credential Framework

Nobutaka KAWAGUCHI†, Naohiro OBATA†, Reina MIYAJI†, Shintaro UEDA†, Hiroshi SHIGENO†, and Kenichi OKADA†

† Faculty of Science and Technology, keio University 3-14-1, hiyoshi, kouhokuku, yokohama, 223-8522 Japan
E-mail: †{kawaguti,obata,miyaji,ueda,shigeno,okada}@mos.ics.keio.ac.jp

Abstract　Credential is a digital value as typified by digital money and digital ticket. In this paper, we propose a P2P credential framework for the use of various type credentials for online and offline. Additionally, we provide a community framework to delegate credential securely online.
Key words　Credential, Peer to Peer, smart card, delegation

1. はじめに

クレデンシャルは様々な個人・組織により発行される電子的価値である．ISP等のサービスプロバイダは会員へのサービスの一貫として電子マネーや有料コンテンツの会員証といった形態のクレデンシャルを発行し，又コンサートやスポーツの興行者は，電子チケットという形態のクレデンシャルを発行する．また，商店のWEBページで発行されたクレデンシャルを現実の商店で使うというオンライン・オフラインを合わせたハイブリッド型や，地域振興や一個人が自身が属するコミュニティに対して発行するローカルマネー[1]といったクレデンシャルも存在する．

筆者らはこれまでスマートカードを用いたP2P環境での協調作業のサポートについて研究してきた[2]．本稿ではスマートカードを利用してピュアP2P環境とオフライン環境の両方に適用可能なクレデンシャルの発行・譲渡・委譲をサポートするフレームワークを提案する．また，P2P環境においてクレデンシャルをネットワーク上に保管するための共同体フレームワークを提案する．

以下，本論文は第2章でクレデンシャルの管理と譲渡の形態について，第3章でP2Pクレデンシャルフレームワークについて，第4章でクレデンシャルの委譲，第5章で共同体フレームワークについて述べ第6章でまとめという構成である．

2. クレデンシャルの管理と譲渡

2.1 クレデンシャルの管理手法

クレデンシャルのフレームワークはその管理手法から以下の3つに分類することができる．

（1）**発行者管理型**　発行者が用意した管理サーバがクレデンシャルを管理する．

（2）**第3者管理型**　使用者が独自に契約を結んだ管理サーバがクレデンシャルを管理する．

（3）**使用者管理型**　使用者自身がクレデンシャルを管理する．

(1)の手法は広く用いられているが，発行者がサーバを設置する必要があり，使用される規模が小さいローカルマネーを発行する際には大きな負担となる．また，オフライン使用ができない．

(2)の手法[3]では発行者の負担はないが，使用者が保管サー

バと契約する費用を払わなければならない．また，クレデンシャルを譲渡する場合，譲渡者と被譲渡者の契約している保管サーバが異なる場合，手続きが複雑となる．またオフライン使用ができない．このように(1)(2)は，想定されるクレデンシャルの使用規模が小さい場合，オンライン・オフラインの両方での使用が望まれる場合は相応しい手法と言えない．

対して(3)では，使用者自身がクレデンシャルを管理する．このため，保管場所としてサーバを用意する必要が無いため，資金的な余裕のない組織や個人が独自にクレデンシャルを発行することができるという利点がある．また，オンライン・オフラインの両方で使用することも可能となる．なお，クレデンシャルの保存媒体としてはスマートカードなどのICカードを用い，クレデンシャルの不正コピーや改竄を防ぐ必要がある．

2.2 クレデンシャルの譲渡

クレデンシャルの譲渡には，(1) クライアント・サーバ型 (C/S型)，(2) P2P型の2通りが考えられる．

C/S型では仲介者を介して譲渡が行われる．この方法は発行者管理型，第3者管理型で用いられる．仲介者は発行者管理型の場合クレデンシャル発行者であり第3者管理型の場合，保管サーバの管理者である．クレデンシャルは譲渡者の保管サーバから被譲渡者の保管サーバへ移動する．

一方P2P型では譲渡者から被譲渡者へ直接譲渡される．この場合，クレデンシャルは譲渡者のスマートカードから被譲渡者のスマートカードへ移動する．この方法は使用者管理型で用いられる．P2P型での譲渡は，現実での通貨やチケットの譲渡と同様であり，自然な形と言える．

3. クレデンシャルフレームワークの提案

自由度の高いP2Pクレデンシャルフレームワークを提案する．ここで自由度が高いとは，様々な形態のクレデンシャルの多様な利用が可能であるという意味である．そのために本提案では以下のことを実現する．

（1） **クレデンシャル発行者のコストの低減** クレデンシャルの発行に際してのコストを最小限に抑え，ローカルマネーなどの小規模なクレデンシャルの発行を可能とする．

（2） **オンライン・オフラインでの使用** オンライン・オフライン両方での使用を可能にする．

（3） **クレデンシャルへのアクセスコントロール** クレデンシャルを譲渡する際にアクセスコントロール機能を付加することにより，譲渡者が使用方法を規定できるようにする．

（4） **複数の種類のクレデンシャルの保持** クレデンシャルは，種類によって独自の性質を持っている．複数の種類のクレデンシャルを一元的に管理することにより利便性を向上させる．

（5） **クレデンシャルの委譲** 一時的なクレデンシャルの委譲を可能とする．このとき，委譲実施者が任意に，譲渡先からクレデンシャル取り戻すことができる必要がある．

（6） **評価の取得** P2Pネットワークでは，今まで面識の無かったユーザとコミュニケーションを取ること多い．あらかじめ，他のユーザの相手に対する評価を取得できれば，初めての相手とコミュニケーションを取るべきか決めるときに，大きな判断材料となる．本提案ではクレデンシャルの委譲の際での利用が考えられる．

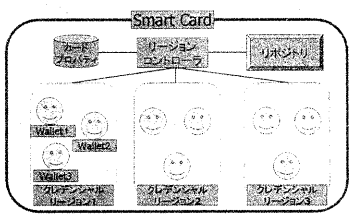

図1 構成要素
Fig. 1 constitent element

上記の事項を実現するために，本提案では使用者管理型のフレームワークを構築する．フレームワークでは，スマートカード内にクレデンシャルを保管する．これにより，発行者のコストを低減し，オンライン，オフラインでの使用が可能となる．

3.1 構成要素

図1にスマートカード内の構成要素を示すとともに，以下に説明する．

（1） **クレデンシャルリージョン** クレデンシャルリージョンは，クレデンシャルの発行者により発行され，1種類以上のクレデンシャルの特性を定義する．また，クレデンシャルに対して最高権限でのアクセスコントロールを行う．

（2） **Wallet** Walletは，内部にクレデンシャルを保持する．また，クレデンシャルへのアクセスコントロールを行う．

（3） **リージョンコントローラ** リージョンコントローラは，カード外との通信，クレデンシャルリージョンの管理を行う．

（4） **カードプロパティ** カードプロパティは，カードの秘密鍵などのカード固有の情報を保持する．

（5） **リポジトリ** リポジトリは，カード内のクレデンシャルリージョン，クレデンシャルに関する情報を保持する．

3.2 クレデンシャルリージョン

クレデンシャルリージョンは，クレデンシャル発行者により発行されるアプリケーションである．クレデンシャルリージョンは以下の2つの要素から成り立つ．

- リージョンプロパティ部
- Wallet管理部

リージョンプロパティ部には以下の機能がある．

（1） **クレデンシャルの定義** 自身に属する1種類以上のクレデンシャルについての定義を行う．

（2） **リージョンIDの保持** リージョンIDはクレデンシャル発行者のカードのハッシュ値で識別される．

（3） **最高アクセス権限リスト** クレデンシャル発行者は，クレデンシャルに対するアクセスコントロールを記述する．具体的には，譲渡の禁止や使用用途の制限などが挙げられる．

（4） **イン/アウトバウンドクレデンシャルの関係付け** リージョンへ入力されたクレデンシャルとリージョンから出力されるクレデンシャルの種類が異なる場合，リージョン内部で両者の対応付けを行う必要がある．例えば，スタンプカードなどでは入力は「スタンプ」というクレデンシャルであり，スタンプ

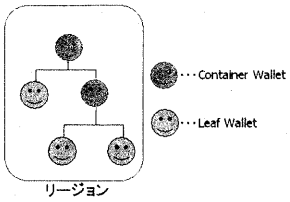

図 2　Wallet の階層構造
Fig. 2　Wallet hierarhcy

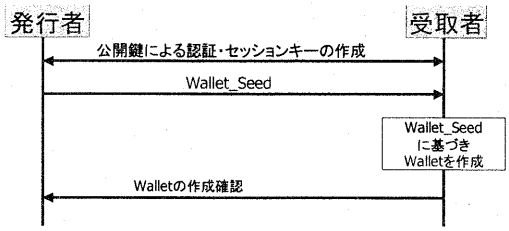

図 3　Wallet の発行
Fig. 3　Wallet publishment

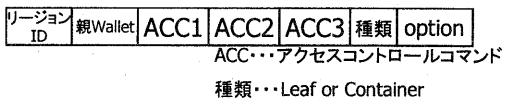

図 4　Wallet_Seed
Fig. 4　Wallet_Seed

が一定数たまると「商品券」というクレデンシャルが出力されることとなる．

Wallet 管理部は，リージョン内に含まれる Wallet 群を管理する．具体的には Wallet の作成，維持，削除を行う．

リージョンの発行者は，そのリージョンに属するクレデンシャルを自由に発行することができる．

3.3　Wallet

Wallet は複数のクレデンシャルを内包する．Wallet は階層構造をとり（図 2 参照），UWI(Universal Wallet Identifer) で識別される．UWI を以下に示す．

UWI = hash(カードの公開鍵) / hash(リージョンの公開鍵) / hash(Wallet 発行者の公開鍵) のパス

具体的には，hash(カードの公開鍵)=alice，hash(リージョンの公開鍵)=yen，hash(Wallet 発行者の公開鍵)=bob であるとき，この Wallet の UWI は

UWI = alice/yen/bob

と表現される．

Wallet には以下の 4 種類がある．

- **LeafWallet** LeafWallet は内部にクレデンシャルと，クレデンシャルに対するアクセスコントロールリストを持つ．
- **ContainerWallet** ContainerWallet は内部に Wallet を保持する．また，アクセスコントロールリストを持ち，内部の Wallet よりも優先度の高いアクセスコントロールを行う．
- **Dele_ProxyWallet** Dele_ProxyWallet は他のカードの Dele_ProxyWallet を参照する．クレデンシャルの委譲の際に用いられる．
- **Dele_RealWallet** Dele_RealWallet は他のカードの Dele_ProxyWallet に参照される．クレデンシャルの委譲の際に用いられる．

3.4　クレデンシャルリージョンの発行

クレデンシャルリージョンを発行するためには，スマートカード内で実行可能なアプリケーションを作成する必要がある．リージョンをアプリケーションで表現することにより，用途に適したクレデンシャルを構築することが可能となる．作成されたアプリケーションは，発行者のカードの秘密鍵で署名され，任意の手段で配布される．リージョンの利用を希望するユーザは，自身のカードにリージョンをインストールする．このとき，リージョンとリージョンの公開鍵のハッシュとの関係が，カード内のリポジトリに記述される．

3.5　Wallet の発行

クレデンシャルリージョンは，そのアクセスコントロールリストの制限に基づいて，リージョン内に Wallet を発行することが可能である．Wallet の発行は，発行対象に基づき 2 通りに分類できる．

- **ローカル発行** Wallet の発行者と受取者が同一
- **リモート発行** Wallet の発行者と受取者が異なる

図 3 に Wallet の発行手順を示す．Wallet_Seed は Wallet を初期化するパラメータであり，アクセスコントロールコマンドを含む．以下では Wallet_Seed とアクセスコントロールコマンドについて説明する．

3.6　Wallet_Seed

Wallet_Seed は Wallet を生成するパラメータである．図 4 に Wallet_Seed の構成を示す．リージョン ID は，この Wallet が作られるリージョンを示す．Wallet オブジェクトは，親 Wallet フィールドに示される Wallet の直下に作成される．指定された親 Wallet が存在しない場合，又はそのフィールドが null を示している場合，Wallet は作成されない．ACC については次節で述べる．

3.7　アクセスコントロール

クレデンシャルの発行者，譲渡者は，発行先や譲渡先に対して，クレデンシャルを使用する上での制限をもうけたいときがある．例えば，顧客が商品を買ったときに発行されるポイントカードが他人に譲渡されることは，発行者としては好ましくない．また，全国チェーンの書店が発行しているポイントカードを母親が息子に譲渡する場合，特定の用途，例えば学習参考書といったものに対して使って欲しいと思うかもしれない．このためには，クレデンシャルの使用用途へのアクセスコントロールを行う必要がある．

アクセスコントロールを行うためにアクセスコントロールコマンド (ACC) を用いる．ACC は，アクセスコントロールリストを構築するエントリーであり，Wallet_Seed に含まれる．ACC の実装の詳細はリージョンに依存するが，基本的には次

UWI	アクション
alice	(<5000;permit) (5000>;ask)
bob	(*;permit)
* (アスタリスク)	(*;deny)

図 5　アクセスコントロールリストの例
Fig. 5　an example of access control list

の 2 要素から構築される．

- ターゲット UWI
- アクション

ターゲット UWI は，アクションの譲渡対象となる UWI を定義する．アクションはクレデンシャルのアクセスコントロール内容を示す．具体的には，クレデンシャルの譲渡を許可する「permit」，譲渡の可否を Wallet 作成者に問い合わせる「ask」，譲渡を禁じる「deny」などのアクションが考えられる．図 5 に ACC から構成されるアクセスコントロールリストの例を示す．この例では，クレデンシャルとして電子マネーを用いている．譲渡先 UWI が alice の場合には，5000 単位以下の場合，permit を，5000 単位以上の場合，ask を実行する．また，アスタリスクなどの特殊記号を使用することにより，ある条件を満たす UWI に一括してアクセスコントロールを行うことができる．

アクセスコントロールリストは，Wallet の UWI のリージョンＩＤ部以降から順に検索される．譲渡先 UWI と該当するターゲット UWI が存在した場合，対応する action が実行され，検索は終了する．該当する UWI が存在しない場合，リージョンごとに規定されたデフォルト処理を行う．

3.8　クレデンシャルの譲渡

クレデンシャルの譲渡の際には，クレデンシャルの ACID 特性を保持する必要がある．つまり，ネットワークやスマートカードのトラブルによりクレデンシャルが紛失したり破壊されることを防ぐ必要がある [4]．図 6 にクレデンシャルの譲渡プロトコルを示す．被譲渡側は，譲渡されたクレデンシャルを保存する Wallet の UWI を申告し，譲渡側はこれに基づき，譲渡のアクセスコントロールを行う．結果として譲渡が許可された場合，2 相コミットでの譲渡が行われる．これにより譲渡セッションが途中で途切れた場合であっても，クレデンシャルの ACID 特性は保持される．

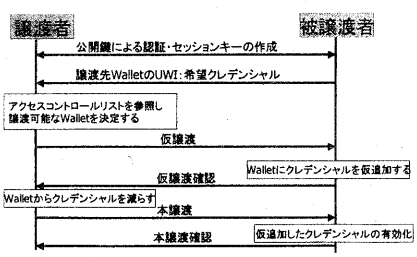

図 6　クレデンシャル譲渡プロトコル
Fig. 6　credential handover protocol

クレデンシャル	相対価値
クレデンシャルA	1.00
クレデンシャルB	0.81
クレデンシャルC	0.52
希望クレデンシャル量:20.00	

図 7　正規化クレデンシャルマップ
Fig. 7　normalized credential map

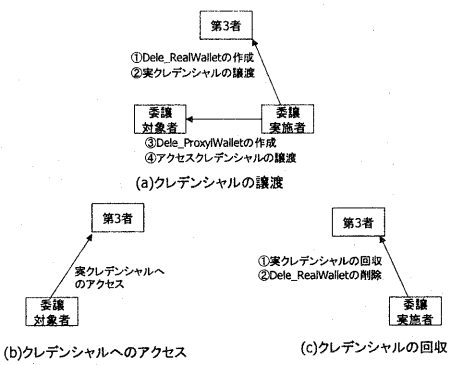

図 8　クレデンシャルの委譲
Fig. 8　credential delegation

3.9　複数のリージョンにわたる譲渡

被譲渡者が複数の種類のクレデンシャルリージョンでの譲渡を望むとき，まず，被譲渡者は譲渡先として希望する Wallet のリストと，正規化クレデンシャルマップを送信する．正規化クレデンシャルマップには図 7 に示すように，被譲渡者にとっての，あるクレデンシャルの基準クレデンシャルとの相対価値と希望譲渡クレデンシャル量が示されている．基準クレデンシャルとは，被譲渡者がもっとも強く譲渡を望むクレデンシャルである．譲渡者はこれに従い，各 Wallet ごとでの譲渡するクレデンシャルとその量を決定する．

4.　オンライン上でのクレデンシャルの委譲

オンライン上でクレデンシャルを委譲する場合，「委譲実施者が任意のタイミングで委譲対象者からクレデンシャルを回収できる」必要がある．委譲時にクレデンシャル自体を委譲対象者のカードに保管することは，以下の点において好ましくない．

- 委譲実施者がクレデンシャルを回収したいタイミングに，委譲対象者がオンライン上に存在しないため回収できない可能性がある．
- 委譲対象者がオンライン上に存在する場合でも，委譲実施者からの回収要求を拒否し不正にクレデンシャルを保持し続ける可能性がある

本フレームワークでは「実クレデンシャルはオンライン上に存在する第 3 者に渡し，委譲対象者にはクレデンシャルに対するアクセスクレデンシャルを与える」ことにより上記の問題を解決する．図 8 にクレデンシャルの委譲，委譲したクレデンシャルへのアクセス，クレデンシャルの回収について示す．実クレデンシャルは Dele_RealWallet 内に，アクセスクレデン

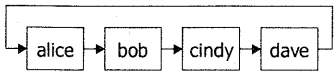

図 9　実クレデンシャルのホップ先リスト
Fig. 9　hop list of real credential

シャルは Dele_ProxyWallet 内に保持される．Dele_RealWallet のアクセスコントロールリストには，アクセスを許可する Dele_ProxyWallet の UWI が記述されている．

委譲実施者，委譲対象者が任意のタイミングでクレデンシャルにアクセス可能であるためには，クレデンシャルは常にオンライン上になければならない．第 3 者に実クレデンシャルを渡したとしてもその第 3 者がオンラインから消えればアクセスは不可能となる．よって，実クレデンシャルはカードからカードへホップすることにより常にオンライン上に存在する必要がある．このため委譲実施者はまず，図 9 に示されるような，実クレデンシャルのホップ先リストを作成する．この例では，実クレデンシャルは alice → bob → cindy → dave → alice の順にホップする．ホップするタイミングには以下の 2 点がある．

- **オフラインになるとき** 実クレデンシャルを保持しているカードがオフラインになる時に，実クレデンシャルを次のカードに渡す
- **委譲対象者からアクセスを受けたとき** 委譲対象者からアクセスを受けたときに，セッション終了時に次のカードに渡す

2 番目の理由は，特定の第 3 者と委譲対象者が結びついて，委譲実施者の回収要求を遮断し，不正に何回もクレデンシャルを使用することを防ぐためである．

クレデンシャルをホップするときに次にホップする予定のユーザがオンライン上に存在せずアクセスが不可能なときには，そのユーザを飛び越えリストの次のユーザへのホップを試みる．しかしこの方法を無制限に行うと，最終的に自分自身にホップして常に実クレデンシャルを持ち続けるという可能性があり，これは上記の理由により相応しくない．このため，規定された回数（最大 jump 長）異なるユーザへのホップを試みて，何れのホップも失敗したならば実クレデンシャルは一時的に無効化されることにする．実クレデンシャルが再び有効化されるのは，後で再びホップを試みて規定回数内で成功した時である．リストが長くなった場合，リスト自体をカードに入れるのは実際的ではない．このため，図 10 に示されるように，リスト自体はネクストホップチップの集合として端末が保持し，ホップコントローラを Dele_RealWallet，Dele_ProxyWallet 内に保持するようにする．ホップコントローラとネクストホップチップに基づき Dele_RealWallet は次のホップ先を調べ，Dele_ProxyWallet は現在の実クレデンシャルの存在場所を検索する．

委譲対象者と実クレデンシャル保持者が結託して委譲実施者からの回収要求を妨害することを難しくするには，以下の方法が考えられる．

- クレデンシャルを分割して複数のユーザに送信する
- ホップリストを作成する際に，信頼度の高いユーザを含めるようにする

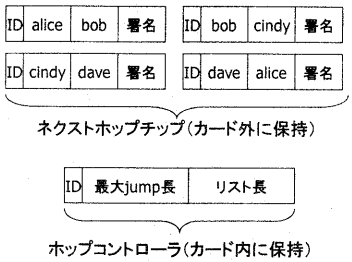

図 10　ネクストホップチップとホップコントローラ
Fig. 10　next hopchip and hop controller

1 番目の方法では，委譲対象者は複数の保持者と結託する必要があるため，不正を働くのが難しくなる．2 番目の手法では不正をする可能性が低いと思われるユーザを選ぶことにより，結託を防ぐ．この信頼度の求め方については次章で述べる．

5. 共同体フレームワークの構築

ネットワーク上でいままで面識がないユーザと初めてコミュニケーションを行う場合，事前にそのユーザへの他のユーザからの評価を知ることで，そのユーザの信頼性を判断することができる．他のユーザからの評価を集約して，あるユーザの評価を行う事に関しては様々な研究がなされている [7] [8]．しかし，これらの研究ではどのユーザもグループや共同体に属さないことを前提としている．このため，各ユーザがあるユーザに対して行う評価は散発的で，その責任性があいまいであり必ずしも信頼性が高いものとは言えない．

本章では，絶対的な Authority が存在しない P2P ネットワークにおける緩やかな共同体フレームワークの構築について考察する．共同体の参加ユーザは，共同体に対して責任を負い，また悪意のあるユーザに対して一致した対応をとる．これにより，より信頼性の高いユーザの評価を行うことが可能となり，前章で述べた，クレデンシャル委譲の際のホップ先を選出する上で大きな判断材料となる．

5.1 共同体の成立

共同体は予めお互いに認識がある複数のユーザにより成立される．共同体の成立にかかわったユーザを Community Core(CC) と呼称し共同体内で最大権限を保持するものとする．また，CC 間で公開鍵の共同体鍵 (CK) を作成する．この鍵により署名された証明書は共同体内で最も強力な証明書となる．

共同体へ他のユーザが加入する際には，加入権限をもつユーザによる認証が必要となる．成立後に参加したユーザを Community Member(CM) と呼称する．図 11 に共同体の概要を示す．

5.2 共同体内の権限

共同体内では様々な権限を定義する．以下に一例を挙げる．

- **加入権限** 任意のユーザを共同体に加入させられる権限
- **追放権限** 共同体内の任意のユーザを共同体から追放できる権限
- **評価権限** 共同体内の任意のユーザに対して評価を行う

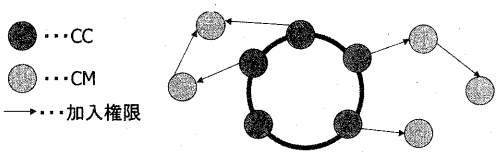

図 11 共 同 体
Fig. 11 community

図 12 証明書の鎖
Fig. 12 credential chain

ことができる権限

- **警告権限** 共同体外の悪意のあるユーザに対する警告を共同体内に行える権限
- **権限付与権限** 上記の 4 権限を任意のユーザに付与できる権限

これらの権限は権限証明書により表現される．あるユーザが正規の権限証明書を保持していることを証明するためには，図 12 に示されるような証明書の鎖 (クレデンシャルチェイン) [5] [6] を構築する必要がある．鎖の先端に位置する証明書は必ず CK により署名されている必要がある．

5.3 証明書の保管

証明書は共同体内のユーザにより保持される．そのため，参照したい証明書を保管しているユーザの IP アドレスを知る必要がある．P2P ネットワーク内でのリソースアドレッシングでは CAN [10] や Chord [9] などが提案されている．証明書を共同体内で保管する場合，悪意のあるユーザが自身が保管している証明書を意図的に破棄したり，また，特定の証明書が頻繁に参照され特定のユーザへの負荷が大きくなる可能性があるという問題がある．これを解決するには，

(1) 証明書の複製

(2) 共同体内での評価が一定以上のユーザによる保持

を行う必要がある．

5.4 ユーザの評価

5.4.1 共同体内のユーザの評価

共同体内でのユーザの評価は CC からそのユーザへの評価の最短パスとなる．これは評価の信頼性は中継されるユーザ数が増えるほど下がるためである．

5.4.2 共同体外のユーザの評価

異なる共同体に属するユーザを評価するには，まずそのユーザが属している共同体を評価する．これは，自身の共同体から相手の共同体への評価の平均値から求めることができる．そして，相手ユーザの共同体内での評価を合わせ，最終的な評価値が定まる．

評価が複数のユーザを中継して遷移する場合，適切な関数により複数の評価を合成する必要がある．例えば，A から B への評価が a，B から C への評価が b である場合，A から C への評価値は eval(a, b) という関数で表せる．この関数には [7] などがある．

6. おわりに

本論文では，スマートカードを用いた P2P クレデンシャルフレームワークを提案した．ネットワークの充実とともに，インターネットショップで取得したクレデンシャルを実世界の店舗で使用したり，個人や小さなコミュニティがクレデンシャルを発行するケースは増加すると考えられる．またクレデンシャルに対し柔軟なアクセスコントロールを行うことで，より利便性を高めることができる．

加えて P2P 環境でのクレデンシャルの委譲をサポートするための共同体フレームワークの構築について考察した．今後は P2P ネットワーク内での共同体の運用やそれに伴うセキュリティ，ユーザ間の負荷分散に関して研究を進めて行きたいと考えている．

文 献

[1] 秋山和隆，並河岳史，手塚一郎，菊池宏徳，山根信二，村山優子， ネットワーク上のローカルマネーシステムの提案
情報処理学会研究報告 2001-CSEC-15 pp317-322 21, Dec 2001.

[2] 小宅宏明，菅原陽子，宮地玲奈，岡田謙一， IC カードを利用したピア・ツー・ピアによるコミュニケーションプラットフォームの提案
情報処理学会研究報告 2002-GN-42 pp13-18 24, Jan 2002.

[3] Kazuo Matsuyama, Ko Fujimura, Distributed Digital-Ticket Management for Rights Trading System
1st ACM Conferences on Electronic Commerce, Nov 1999, pp. 110-118.

[4] Kimio Kuramits, Tadashi Murakami, Hajime Matsuda, Ken Sakamura, TTP:Secure ACID Transfer Protocol for Electronic Ticket between Personal Tamper-proof Devices
The 24th Annual International Computer Software and Application Conference Oct 2000, pp. 87-92.

[5] Eric Freudenthal,Tracy Pesin,Lawrence Port,Edward Keenan, Vijary Karamcheti, dRBAC:Distributed Role-based Access Control for Dynamic Coalition Environments
Proceedings of the 22nd International Conference on Distributed Computing Systems, 2002.

[6] Ting Yu, Marianne Winslett, Kent E. Seamons, Supporting Structured Credentials and Sensitive Policies through Interoperable Strategies for Automated Trust Negotiation
ACM Transactions on Information and System Security, Vol. 6, No. 1, Feb 2003, pp. 1-42.

[7] Bin Yu, Munindar P. Singh, A Social Mechanism of Reputation Management in Electronic Communities
Proceedings of Fourth International Workshop on Cooperative Information Agents, 2000, pp. 154-165.

[8] L. Xiong, L. Liu, Building Trust in Decentralized Peer-to-Peer Electronic Communities
Proceedings of the Fifth International Conference on Electronic Commerce Research, Oct, 2002

[9] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, Hari Balakrishnan, Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications
Proceedings of ACM SIGCOMM 2001

[10] Sylvia Ratnasamy , Paul Francis, Mark Handley, Richard Karp, Scott Shenker, A Scalable Content-Addressable Network
Proceedings of ACM SIGCOMM 2001.