Vol.2015-CSEC-69 No.30
Vol.2015-IOT-29 No.30
2015/5/22

UPKI パス : FCF キャンパスカードと証明書ストアサーバとの連携によるクライアント証明書活用システムの改良

中村素典[†1]　西村健[†1]　山地一禎[†1]

認証処理におけるパスワードの脆弱性が強く認識されるようになり，より強力な認証手段が求められている．そのような認証手段の 1 つとして PKI に基づくクライアント証明書が有効であることが広く知られているが，クライアント証明書の利用者側の取り扱いが煩雑であることから十分な普及には至っていない．一方，大学では，FeliCa カードをベースとした FCF キャンパスカードが学生証や教職員証として広く普及し認証にも利用されている．しかし，FCF キャンパスカードの標準的な利用形態においては，PKI クライアント証明書の秘密鍵をカードに保存するための領域を確保することができず，また，TPM や HSM のような，秘密鍵を取り出せないようにしつつ暗号化や復号の処理を可能とする機能も備えていないため，そのままではクライアント証明書と組み合わせた利用が困難である．このような FCF キャンパスカードにおいて，PKI クライアント証明書活用する手段の一つとして，証明書ストアサーバとの連携による JCAN パスが提案されている．本稿では，この JCAN パスのセキュリティを向上させ，UPKI クライアント証明書の発行形態に対応させた UPKI パスについて述べる．

UPKI-Pass – An Improvement of the System Combining FCF Campus Cards and A Certification Storing Server to Utilize PKI Client Certificates

MOTONORI NAKAMURA[†1]　TAKESHI NISHIMURA[†1]
KAZUTSUNA YAMAJI[†1]

Stronger authentication methods should be used since weakness of password authentication has been widely known. Client certificates based on public key encryption mechanism is widely known as one of stronger authentication methods. But, it has not been widely deployed because of difficulties of its management at user side and costs. By the way, Smart Card based on FCF Campus Card standard is widely deployed in universities in Japan as Student certificates and/or Staff/Faculty member certificates. But the FCF Campus card format does not have an area to store a pair of keys (secret key and public key) in it, and it does not have mechanism for tamper resistance like TMP or HSM to protect secret keys not to be picked out. So it is difficult to utilize client certificates with FCF Campus Cards. To overcome such difficulty, a mechanism called “JCAN Pass” has been proposed to utilize Client Certificates with FCF Campus Cards by adding a Certificates Storing Server. In this report, a new mechanism called UPKI Pass is proposed. UPKI Pass improves security of “JCAN Pass” and fits to issuing procedure for UPKI client certificates.

1. はじめに

社会システムの電子化，オンライン化の進展において，利用者の識別を行う認証の役割は非常に重要である．古くから利用されてきたパスワードによる認証には様々な脆弱性が指摘されており，より強固な認証手段への移行が喫緊の課題となっている．

より強固な認証手段の 1 つとして公開鍵基盤（PKI; Public Key Infrastructure）に基づくクライアント証明書の利用がある．しかし，発行の手順が複雑で利用者自身が注意深く扱わなければセキュリティの低下を招きやすい，発行コストがかかりすぎる等の理由により，その普及はあまり進んでいない．クライアント証明書の取り扱いが容易な，暗号処理に対応し耐タンパ性を持つ IC カード（Smart Card とも呼ばれるが，ISO/IEC 7816 に準拠する接触仕様 JavaCard や ISO/IEC 14443 に準拠する非接触仕様 Type B など）や USB トークンを活用する方法もあるが，やはり証明書発行や更新時の手間と，デバイスのコスト高により幅広い普及に向けた基盤環境とはなり得ていない．

このような状況の中，一般財団法人日本情報経済社会推進協会(2011 年に財団法人日本情報処理開発協会から改称. 以下，JIPDEC)では，広く普及している IC カードの 1 つである FeliCa [1]を活用しクライアント証明書の扱いを簡便にする JCAN パス[2][3]を提案している．FeliCa は耐タンパ性を備えつつ公開鍵暗号を処理する機能を持つものではないが，サーバから暗号化された秘密鍵を取得し，カードに保存された鍵を用いて復号するという方式を採ることで，クライアント証明書が簡便に利用可能となる．

JCAN パスは，その仕様により，耐タンパ性を持つ IC カードと比較すれば秘密鍵の扱いに関するセキュリティレベルは低くなるものの，一般的なクライアント証明書の利用形態としては十分実用可能であると考えられる．しかし，認証基盤として広く普及させるためには，その仕様の細部にわたる十分なセキュリティ上の検討が求められる．そこ

†1 国立情報学研究所
National Institute of Informatics

で，本稿では，JCAN パスのセキュリティを向上させるための検討を行うとともに，その改良版である UPKI パスの概要について述べる．

2. クライアント証明書普及に向けた取り組み

2.1 クライアント証明書

通信の保護や認証等に用いられる情報セキュリティ基盤の一つである Public Key Infrastructure (PKI)は，1998 年に初版が公開された X.509[4]において電子証明書の形式とその発行のための認証局の階層構造などが定義され，それはインターネットにおける標準にも引き継がれ広く活用されている[5][6][7]．電子証明書は，いわゆるクライアント―サーバモデルにおいて，サーバ側で用いられるサーバ証明書と，クライアント側で用いられるクライアント証明書に大別されるが，いずれについても，取り扱いが煩雑であり，秘密鍵を危殆化させないよう注意深く管理し信頼性を維持する必要がある．サーバ証明書は一定以上の知識と技術を持つと期待されるサーバ管理者によって管理されるため，信頼性の維持がさほど困難ではないが，クライアント証明書はその利用形態によってはセキュリティレベルが一般利用者の管理能力に大きく依存するため，セキュリティレベルを保った展開が容易でない．

クライアント証明書の扱いを容易にするための方法として IC カードや USB トークンなどの耐タンパ性のあるデバイスで秘密鍵を保持し利用する方法が登場したが，デバイスやリーダライタ（IC カードの読み取りと書き込みを行う装置）のコストが小さくないことや，証明書更新時のデバイスの回収と再配付等に係る管理コストもかかることから，その普及は限定的である（大学においては，教職員のみに配付し，学生にまでは配付しない事例が少なくない）．

また，クライアント証明書の発行コスト削減のため，パブリックな証明書を用いる代わりに，CA（認証局; Certification Authority）を組織内に構築してプライベートな証明書を発行する形態を採っている大学等もある．組織内での認証用途に限定すれば，パブリックな証明書を利用する必要はないが，プライベートな証明書であっても，一定以上のセキュリティ要件を維持した発行管理を行うためには，そのコストも決して安くはない．従って，各大学が独自の CA を構築しプライベートな証明書を発行する方法は，全ての大学等に展開可能な形態とは考えられない．

ちなみに，電子メールの暗号化や電子署名を行う S/MIME [8]もパブリックなクライアント証明書を利用する魅力的なアプリケーションの一つであると考えられるが，同様に証明書管理の煩雑性からあまり普及には至っていない．

2.2 JCAN 証明書

JIPDEC では，普及の妨げとなっているクライアント証明書の発行コストや発行手続きの煩わしさの解消を目的として JCAN (Japan CA Network)と呼ぶ枠組みに基づく実証実験を 2011 年より開始している．JCAN ではサイバーID 証明書 JCAN と呼ばれるクライアント証明書が発行されているが， JCAN 認証局から，参加組織の構成員に対してパブリックなクライアント証明書が発行される仕組みである．各参加組織に証明書の発行権限を持つ LRA (Local Registration Authority)を置き，各 LRA が認証局業務の一部を代行することで，というモデルを採ることで証明書発行の低価格化と発行の煩雑さの解消を実現している[9].

さらに，電子証明書にはドメイン名と組織名称程度の情報しか記載されておらず，証明書から得られる情報のみでは当該組織の信頼性に関する十分な情報が得られないという問題点を補うことを目的として，JIPDEC ではサイバー基本台帳 ROBINS と呼ばれるサービスも 2012 年より開始している [10] ．

2.3 JCAN パス

JCAN 証明書を活用するアプリケーションの一つとして JCAN パス（JCAN Pass）と呼ばれる仕組みが JIPDEC に設置されたワーキンググループの下、福田昭和氏を中心に 2010 年より検討が開始された（当初は JCAN ビジネスパスと呼ばれていた）[11].「パス」は通行許可証の意であり，PC 等の端末と IC カードを組み合わせて利用するしくみである．

JCAN パスでは FeliCa を用いる．FeliCa は日本国内において IC カード乗車券や電子マネーとして広く普及が進んでおり，カードやリーダライタのコストは十分下がっている．企業や大学等でも身分証や入退カードとしても利用が進んでいる．

FeliCa を身分証や入退カードとして利用する場合は，複数のサービスに容易に対応が可能なマルチユース用途向け規格である FCF （Felicitous Common Format, 共通利用フォーマット）が広く採用されており，さらに大学等では，教育機関向けの統一規格である FCF キャンパスカードの導入が進んでいる．JCAN パスも，FCF（キャンパスカードを含む)に準拠している.(以下では, 特に断りがない場合, IC カードとは，FCF に準拠した FeliCa のことを指す.)

FCF では，クライアント証明書のような大きなデータを IC カードに格納することは想定されていない．そこで，JCAN パスでは，クライアント証明書および秘密鍵自体を IC カードの中に格納する代わりに, 解凍パスフレーズを IC カードの中に保持する．クライアント証明書および秘密鍵は，多くの場合その交換に PKCS#12 形式 [12]が用いられるが, PKCS#12 では原則として暗号化された秘密鍵が格納され, 解凍パスフレーズは PKCS#12 を復号する際の鍵となる．解凍パスフレーズはクライアント証明書や秘密鍵に比べて十分小さいため, IC カードに容易に格納することが可能である．

PKCS#12 は 3DES 等の十分強力な暗号スイートを用いて保護しつつ, IC カードに格納された解凍パスフレーズを利

Vol.2015-CSEC-69 No.30
Vol.2015-IOT-29 No.30
2015/5/22

用者の目に触れさせないように機械処理させることにより，利用者の不注意による秘密鍵の危殆化を防いでいる．JCAN パスでは，PC 等の端末での利用を前提としており，IC カードをリーダライタにセットしている間のみ，復号された秘密鍵とクライアント証明書が端末上の証明書ストアにインストールされ利用可能となる．IC カードをリーダライタから外すと，秘密鍵とクライアント証明書は端末の証明書ストアから消去される．なお，IC カードから解凍パスフレーズを読み出す際には，IC カード PIN 照合による本人認証も要求することで，単純な盗難では不正利用されない二要素認証を実現している．

解凍パスフレーズと対となる PKCS#12 は別途取得する必要があるが，その取得方法について，JCAN パスではこれまで次の 3 段階で発展してきている．

1. ローカルファイル型
   暗号化された PKCS#12 形式のファイルを利用者に配付し端末上の決められた場所に保存し，FeliCa アプリ―ケーションから参照．
2. ローカルサーバ型
   1 と同様だが，FeliCa アプリケーションが PKCS#12 形式のファイルを参照する際に，ローカルに閉じたクライアント―サーバ通信を利用する．
3. リモートサーバ型
   FeliCa アプリケーションは，定められたサーバにアクセスして PKCS#12 形式のファイルを取得

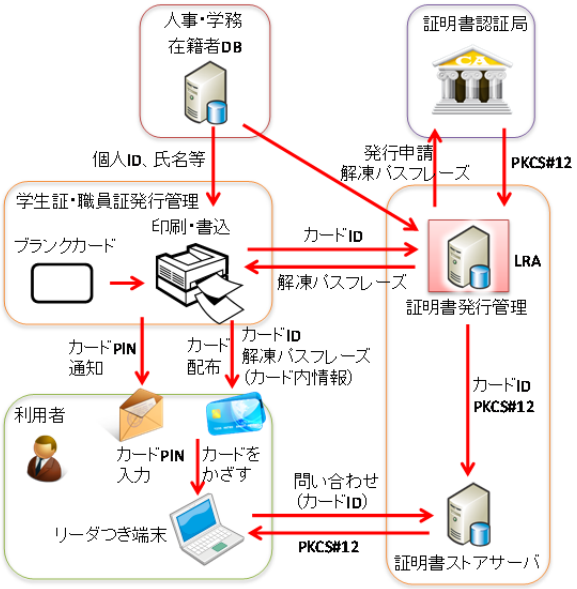

図 1. JCAN パスシステムの概念図

最新の JCAN パスでは，最後のリモートサーバ型を採用しており（図 1），本稿でも，リモートサーバ型を前提として議論を進めるが，リモートサーバ型が実現されたことにより，以下のようなメリットに結びついている．

- 利用者は PKCS#12 形式ファイルのことを一切意識する必要がない
- サーバ側単独で証明書の更新が可能
- 証明書のより迅速な無効化処理が可能（特に、IC カードを紛失した場合において、Certificate Revocation List (CRL)への登録による証明書の失効処理を待つ必要がない）

また，JCAN パスでは，複数の解凍パスフレーズをカードに格納することで，複数の証明書を同時に扱うことができる．複数の証明書が扱えることで，以下のような場合にも 1 枚のカードで対応が可能である．

- 古い証明書の継続的な保持
  S/MIME のような電子メールを暗号化して送受信するアプリケーションでは，過去に受け取った暗号化メールを復号するために期限が切れた秘密鍵も保持し続ける必要がある．
- 複数ロールへの対応
  ある利用者が組織内で複数の役割を持つような場合（大学院生かつティーチングアシスタントなど）は珍しくない．そのとき，どの役割としてのアクセスであるかを認証によって判断するようにシステムが設計されていることも多く，そのような場合に複数の証明書の使い分けが必要となる．
- グループ認証への対応
  特定のグループ内でのファイル共有などの際に，そのグループへの所属を証明書に基づいて判定するようなアプリケーションも考えられる．どのグループに所属しているかを記載した証明書を利用者ごとに発行することによって，秘密鍵のグループ内での共有を行わずに，グループアクセスを実現することが可能であり，このようなアプリケーションを利用する際には，複数の証明書の使い分けが必要となる．

なお，最新の JCAN パスは，2013 年 11 月に公開された FCF バージョン 3 [13]（表 1）の利用を前提としており，最大 8 つまでのクライアント証明書に対応することが可能である．ちなみに，FCF バージョン 3 より古い IC カードを FCF バージョン 3 に更新するためには，IC カード製造（1 次発行）を行う工場での作業が必要となることから，基本的に IC カードの再発行による対応となる．

表 1. FCF バージョン 3 で定義されるエリアの概要

エリア	用途	読出鍵	書込鍵	ブロック数
システム	製造ID			4
A	基本ID情報	なし	あり	8
N	FCF-UN	なし･あり	あり	6
B	追加サービス履歴	あり	あり	10
C1	追加サービス	あり	あり	7
C2	追加サービス	あり	あり	7
C3	追加サービス	あり	あり	13
C4	追加サービス	なし	あり	7
D1	追加サービス	なし	なし	16

（文献[2]を基にバージョン 3 に関する情報を追加）

表 2.　6 つの課題

1	UPKI 共通仕様の作成・配布
2	オープンドメインサーバ証明書発行
3	大学間無線 LAN ローミング
4	シングルサインオン検討
5	認証局ソフトウェアパッケージ開発
6	S/MIME 証明書の試験利用

2.4 UPKI 電子証明書発行

国立情報学研究所（以下 NII）では，大学等がインターネットを活用して提供する各種オンラインサービスのセキュリティ向上に向けた取り組み「大学間連携のための全国共同電子認証基盤（UPKI）構築事業」を文部科学省からの支援も得て 2006 年に開始した．本事業では 6 つの課題（表 2）を設定したがそのうちの 1 つとして，Web サービス等へのアクセスの安全性を向上するために必要となるパブリックなサーバ証明書の普及を促進するため,「サーバ証明書発行・導入のための啓発・評価研究プロジェクト」を 2007 年より開始した．同プロジェクトでは，NII がサーバ証明書発行に必要な費用を負担し，大学等は無償でサーバ証明書が入手できるようにすることで普及促進を図ることを狙ったが，発行に係る費用を抑えるため，各大学等にも発行時に必要となる各種確認作業の役割を分担させた学術スキームを採用した[14]．このプロジェクトは後続の「UPKI オープンドメイン証明書自動発行検証プロジェクト」を経て，参加機関数 337（ドメイン数 360），のべ発行枚数 23734（後継プロジェクト分のみで積算，うち有効枚数 11088）というところまで普及するに至った（2015 年 3 月末現在）. 2015 年からは，これまでの実証実験的発行の成果を踏まえ，発行に必要な費用を各大学等に分担頂きつつ NII の正式な事業としてサービスを継続的に提供することとしたが，クライアント証明書およびコードサイン証明書についても，学術スキームの下で発行を開始することとした．クライアント証明書およびコードサイン証明書は当面の間（3 年間を予定）追加の費用負担なしに枚数無制限に発行可能とし，その後有償（発行枚数に寄らない定額制）サービスに移行することを予定している．このような学術向けの有償証明書発行サービスは，欧州 TERENA による TCS [15]や米国 Internet2 の InCommon によるサービス[16]等の先行例がある．

UPKI 電子証明書発行サービスによるクライアント証明書の発行が始まると，多くの大学等で活用に向けた検討が始まると考えられる．UPKI クライアント証明書の展開のためにも，JCAN パスの仕組みは大いに活用できると期待される．

3. UPKI での JCAN パス活用に向けた検討

大学等における JCAN パスの活用可能性を検討するため，UPKI におけるクライアント証明書の発行に先立って，NII において試験的に JCAN パスシステム導入を行った．技術仕様の詳細を確認したところ，次のような 4 つの懸念点が判明した．

3.1 本人認証のためのカード PIN の扱い

IC カードを端末のリーダライタにセットした際，本人による利用であることを確認するために，本人のみが知る知識であるカード PIN の入力が求められる（2 要素認証）．このカード PIN を照合するための情報（最大 16 文字）は IC カード内に格納されているが，カード固有識別子（FeliCa 製造番号）である 8 バイト（16 桁）の IDm [17]と非公開のソルトの組み合わせを鍵として暗号化されているだけであり，IDm 自体はカードから容易に読み出せる情報である．また，暗号化は JCAN パスの他の情報を含めて一括して行われており，フォーマットは FCF 会員組織限定ではあるが公開されていることから，既知平文攻撃が可能であり，一旦 1 枚の IC カードに対する解読が成功すれば，その他の IC カードのカード PIN も容易に解読でき，簡単に本人に成りすませてしまう，という点において脆弱である．

3.2 証明書ストアサーバへのアクセス

端末は，カード PIN が正しいことを確認すると，証明書ストアサーバから暗号化された PKCS#12 形式の鍵対を入手するが，このとき，端末から証明書ストアサーバには要求する PKCS#12 に対応する識別子（IDm 等）を伝えるのみであり，利用者認証は行われていない．IDm は暗号化せずに IC カードに書き込まれている情報であり，一般的なリーダライタを用いて IC カードから容易に読み出すことができる．従って，証明書ストアサーバに対する通信プロトコルが既知で証明書ストアサーバにアクセスできれば容易に PKCS#12 を入手することが可能である．証明書ストアサーバをインターネットに接続せず，組織内からのみアクセス可能とすることでセキュリティを確保する対策もあり得るが，インターネット上で提供される様々なサービスを ID

連携によるシングルサインオン環境で活用していこうという時代の流れの中において，パスワードの代替としてクライアント証明書を用いた認証が求められるようになれば，世界中どこにいても JCAN パスシステムが利用できるべきであり，証明書ストアサーバでの利用者認証は必要不可欠である．

3.3 端末上の証明書ストア自身のセキュリティ

Windows 端末上の証明書ストアに一時的に格納される秘密鍵は，利用者による証明書ストアからの取り出しを防止するため，取り出しを不可とする設定となっている．しかし，その設定によらずシステムから強制的に取り出すツールやウィルスが存在している．また MacOS ではそもそも証明書の取り出しを禁止することができないため，利用者が容易に転用できてしまい，セキュリティレベルの低下が懸念される．従って，これらに対する対策が求められる．

3.4 証明書発行形態の違い

PKCS#12 形式の秘密鍵と公開鍵証明書の鍵対は証明書ストアサーバから端末に送られ，復号されて取り出された鍵対が端末の証明書ストアに格納される．この PKCS#12 の復号時に必要となる解凍パスフレーズは，PKCS#12 を暗号化する際にも用いられる共有鍵であり，CA がクライアント証明書を発行する際にも必要である．JCAN パスシステム設計時の前提である JCAN 証明書では，事前に生成した解凍パスフレーズを CA に通知し，通知した解凍フレーズによって暗号化された PKCS#12 を受け取る，という処理の流れとなっている．このような場合，CA から発行された PKCS#12 は証明書ストアサーバを経て端末に送られるまでに解凍フレーズと同じ場所に置かれることがないため，証明書ストアサーバが侵入を受けたとしても，すぐさま秘密鍵が危殆化するわけではなく，悪用されるまでに鍵を失効させる時間が確保できる．しかしながら，PKCS#12 を暗号化する解凍フレーズを事前に指定できるかどうかは CA の運用に依存するものであり，UPKI 証明書発行サービスにおいては，委託先となる証明書発行業者の仕様により事前に指定することができない．そのため，同時に発行される PKCS#12 および解凍フレーズを安全に扱う処理フローを設計する必要がある．

4. UPKI パス方式の提案

前項で示した JCAN パスシステムの 4 つの懸念点を解決するための方法を以下に述べる．これらの方法に基づき改良したものを UPKI パス方式と呼び，UPKI パス方式に基づくシステムを UPKI パスシステムと呼ぶ（図 2）．UPKI パス方式では，IC カードに係わる発行・運用管理コストの低減についても併せて考慮した．

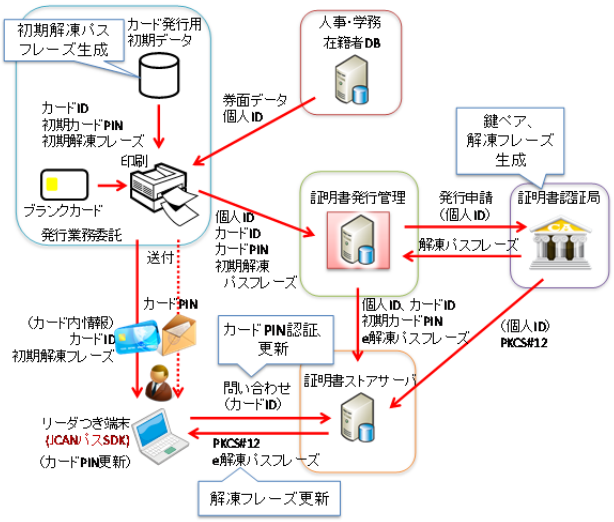

図 2. UPKI パスシステムの概要

4.1 IC カード識別子と暗号鍵

JCAN パスでは，FCF バージョン 3 で定義される C4 エリアおよび D1 エリアに，解凍パスフレーズやカード PIN などのデータを暗号化したものを書き込んで利用するが，暗号化の際には FeliCa における IC カード固有の識別子である 16 桁の IDm が鍵の一部として用いられる．これは，データの秘匿に加えて IC カードの単純複製の防止の役割を持つ．IDm は Sony の管理の下，IC カード製造（1 次発行）時に書き込まれ，その後の書き換えが不可能であることから，（1 次発行工程の信頼性が損なわれない限り）一般的に入手可能な IDm 書き込み済みの別カードに単純複製したとしても，C4/D1 エリアは復号に失敗する．

しかし，上記の IDm は一部のエミュレータにおいて容易に設定できる等の理由により，高セキュリティが要求される処理においては IDm のみに依存した処理は推奨されておらず，FCF バージョン 3 では，IDm に代わる別の識別子として新たに FCF-UN が定義されている[18]．FCF-UN も IDm と同様に 1 次発行時に管理され，その後の書き換えは不可能である（FCF-UN は券面にも記載される）．FCF-UN は読み出し時に暗号化をしなければ（表 1 の「読出鍵」を「あり」にしてのアクセス），IDm とセキュリティ的には大きな差はないが，FCF では IC カード識別子として 16 桁（8 バイト）の FCF-UN の使用を推奨していることから，UPKI パスでは FCF-UN を鍵の一部として D1 エリアの暗号化に用いる．（4.4 節で後述の通り，C4 エリアは UPKI パスでは使用しない．）

さらに，IC カードと証明書ストアサーバの管理は組織毎に行われ，両者の関係が一意に定まることを利用して，FCF-UN に加えて組織毎に異なる値を D1 エリアの暗号化に利用することで，さらにセキュリティを向上させることが可能である（一部の組織の IC カードで暗号鍵の値が明らかになっても，直ちに他の組織の運用に影響しない）．

4.2 本人認証用カード PIN

IC カードに書き込まれたカード PIN の読み出しと解読を防止する単純かつ確実な方法は，カード PIN を IC カードに書き込まないことである．オフライン環境においてカード PIN を用いた本人確認を行う必要がなければ，カード PIN を IC カードに書き込む代わりに，組織毎に用意した認証サーバに対して認証を行うことで本人確認を行うことが可能である．サーバに対して認証することで，総当たり攻撃による解読を回避することが可能となる．

IC カードには，カード PIN の他にも PKCS#12 を復号するための解凍パスフレーズも暗号化して格納されている．この解凍パスフレーズは，IC カードを入手して総当たり攻撃等を行うことで解読することが可能であるが，解凍パスフレーズが入手できたとしても対応する PKCS#12 が入手できない限りすぐさま安全性に支障を来すことはない．IC カードの紛失に気づいた時点で証明書ストアサーバ上の PKCS#12 を無効とし，IC カードを含めた再発行処理を行えば，不正アクセスを許すことはない．

4.3 証明書ストアサーバとの認証

インターネット上の様々なアプリケーションにおけるクライアント証明書の活用を支援するためには，インターネット上の端末において利用者の IC カードに紐付いた PKCS#12 を証明書ストアサーバから取得できるようにする必要がある．その際，第 3 者によって PKCS#12 が容易に取得されないように証明書ストアサーバにおいて認証を行う必要がある．証明書ストアサーバが前節で述べたカード PIN を検証する認証サーバの役割を担うことで，利用者の本人認証と同時に PKCS#12 のアクセス認可を行うことが可能となる．

認証サーバによる認証では，認証情報の漏洩を起こさないように認証サーバの成りすまし対策が不可欠である．そのためには，端末において認証サーバの正当性を検証する必要がある．端末と認証サーバの通信には，十分強力なサーバ証明書と暗号スイートを用いた TLS 通信で保護するとともに，MS-CHAPv2 [19]等の相互認証に対応した認証方式を用いることで，端末側からも認証サーバ（具体的には証明書ストアサーバ）の真正性を検証する．（MS-CHAPv2 は、TLS で保護すれば直ちに問題とはならないが脆弱性が指摘されていることに注意が必要である．）

認証に用いるカード PIN は利用者による変更に対応する必要があるが，そのためには証明書ストアサーバと連携した処理が必要となる．よって，カード PIN 変更のプロトコルを新たに定義する．

4.4 解凍パスフレーズ更新管理

UPKI 電子証明書発行サービスでは，PKCS#12 の解凍パスフレーズを利用者側で事前に生成し，鍵対の入った PKCS#12 の暗号化をその解凍フレーズを用いて行うように指定することができない．代わりに，CA が解凍パスフレーズを生成し，それを用いて暗号化された PKCS#12 とともに受け取るという仕様である．ただし，受け取り時の脆弱性を回避するため，これらは別個に，異なる担当者が異なる手段を用いて受け取る配慮がなされている．UPKI パスシステムにおいても，PKCS#12 と，それを復号するための解凍パスフレーズが同時に同じサーバ上に存在しないような配慮が求められる．同一サーバに両者が同時に存在しなければ，一方のサーバが侵入を受けたとしても，直ちに秘密鍵が危殆化し被害が発生することはないと考えられるが，侵入を受けたら直ちに鍵の失効等の必要な対策が開始できるような監視，運用体制を構築しておくことが重要である．

解凍パスフレーズは IC カードへの書き込みが必要であるが，前述のように，UPKI 電子証明書発行サービスでは証明書発行前に書き込まれた解凍パスフレーズによって暗号化された PKCS#12 を入手することができず，証明書発行後に IC カードに書き込む必要がある．また，学生証として利用する場合，クライアント証明書の有効期限は最大 3 年となっており（電子証明書発行サービスの委託契約期間による制約），これは学部生の在学年限である 4 年より短いことから，必ず 1 回の証明書更新処理が発生することになる．その際にも，解凍パスフレーズの更新が必要となる．

JCAN パスでは，IC カード発行時に書き込んだ解凍パスフレーズをそのまま固定的に使用するという仕様であったが，UPKI パスでは，証明書発行後に IC カードの解凍パスフレーズを書き換えることとした．具体的には，IC カードの初回アクセス時と，証明書の更新時に書き換えを行うこととなる．

最新の JCAN パスは，2013 年 11 月に公開された FCF バージョン 3 の利用を前提としており，C4 エリアと D1 エリアと呼ばれる 2 つのエリアを組み合わせて利用している．D1 エリアはバージョン 3 において新たに定義されたエリアである．C4 エリアの書き込みには専用のリーダライタが必要であるが，C4 エリアの読み出しと D1 エリアの読み書きは汎用リーダライタで可能である．C4 エリアには，PKCS#12 を復号するための解凍パスフレーズを 2 つ分保持することができ，D1 エリアには解凍フレーズを 6 つ分保持することができる．すなわち，JCAN パスでは IC カード全体で 8 つ分の解凍フレーズを保持することができ，同時に 8 つのクライアント証明書を扱うことが可能となっている．

一方，UPKI パスでは，解凍フレーズの書き換えが必要であり，C4 エリアの解凍パスフレーズを書き換えるためには，専用のリーダライタを設置した窓口等において対応する必要が生じる．IC カードの初回利用時にも書き換えが必要となることから，汎用リーダライタで必要に応じて書き換え対応が可能となるように，UPKI パスでは C4 エリアを使用しないこととした．

なお，IC カード上には解凍パスフレーズ 1 つにつき 16

バイト分の領域が確保され，最大 128 bit の鍵を扱うことができる．解凍パスフレーズとして，一般には 12 桁が広く用いられているが，ほとんどのアプリケーションは 16 桁に対応しているため，UPKI 電子証明書発行サービスでは，16 桁の解凍フレーズを生成することとしている．16 桁といっても，利用可能な文字は限定されているため，実際には 96 bit 相当である．将来的により長い，すなわち，より強力な解凍パスフレーズも扱えるようにするため，UPKI パスでは圧縮して保持するように設計し，21 桁（126 bit 相当）まで扱うことが可能としている．

4.5 端末における証明書管理

JCAN パスには，Windows および Mac 端末に対応したソフトウェアが存在しているが，これらは OS の持つ証明書ストアに秘密鍵と公開鍵証明書の鍵対を一時的にインストールする実装となっている．すなわち，IC カードがリーダライタにセットされ本人認証に成功し PKCS#12 を証明書ストアサーバから取得すると，復号した後 OS の証明書ストアに鍵対がインストールされ，IC カードがリーダライタから外されると鍵対が消去される．Windows の場合は，OS が提供する標準インタフェースを用いて利用者が鍵対を取り出せないように設定できる機能があるが，MacOS には，そのような機能が存在しないため，IC カードがリーダライタにセットされている間に容易に鍵対を取り出すことが可能である．また，Windows についても，設定にかかわらず不正に証明書ストアから鍵対を取り出すことができるとされており，ウィルス感染等による危殆化が懸念される．

Windows や MacOS では，鍵対を OS の証明書に保持する以外にも，耐タンパ性のある証明書デバイスを用いるための PKCS#11[20]や CAPI/CNG[21]等のインタフェースが用意されている．UPKI パスをこれらのインタフェースを介した処理に対応させることで，端末上の標準的な証明書ストアからの鍵対の取り出しへの対策を講じる．

5. UPKI パスシステムの実装

3 章で示した方針に基づき，UPKI パスシステムを実装した．IC カードに対する処理の流れを以下に示す．なお，これは IC カード発行処理の手間をできるだけ削減するように配慮を行ったものとなっている．

1. IC カード製造・納品
   IC カード製造（1 次発行）時にランダムな初期解凍パスフレーズ（6 つ分）をカードに書き込み，IC カードと一緒に納品（UCF-UN と初期解凍パスフレーズとの対応リストも同時に納品）．必要に応じて，納品後に券面印刷と学籍番号等の書き込みを行う（2 次発行）．
2. IC カード配付と初期カード PIN 通知
   初期カード PIN を生成し，FCF-UN と紐付いた対応リストを作成して証明書ストアサーバに格納．本人認証を経て IC カードと初期カード PIN を利用者に配付．
3. 証明書発行
   利用者ごとに必要な枚数のクライアント証明書を発行申請（1000 枚ごとのバルク発行処理）．発行を依頼した証明書の CN (Common Name)と FCF-UN との対応リストを作成．
4. 発行された解凍パスフレーズの受け取り
   受け取った解凍パスフレーズ（以下，新解凍パスフレーズ）と 1 で納品を受けた初期解凍パスフレーズとの対応をとり，新解凍パスフレーズを初期解凍パスフレーズで暗号化した後，FCF-UN と対応をとり，証明書ストアサーバに格納する．（暗号化されていない新解凍パスフレーズを証明書ストアサーバ上に持ち込まないことで安全性を確保する．）
5. 発行された PKCS#12 の受け取り
   受け取った PKCS#12 を証明書ストアサーバに格納（解凍パスフレーズおよび FCF-UN との対応をとる）

（以上で準備が完了）

6. カード PIN による本人認証
   利用者が IC カードを端末のリーダライタにセットすると，カード PIN の入力を求める．証明書ストアサーバに対して TLS 通信路を設定しサーバの真正性を確認した後，FCF-UN および入力されたカード PIN を用いて認証を行う．（認証成功後，初回アクセスの場合はカード PIN の変更を要求する．）
7. 初回アクセス時の解凍パスフレーズ更新
   利用者による初回アクセス時に，証明書ストアサーバは暗号化された新解凍パスフレーズを送信する．受信した端末は，（複数の解凍パスフレーズの対応関係を確認した上で）IC カードから（復号して）取り出した初期解凍パスフレーズを用いて，新解凍パスフレーズを復号し，IC カード上の解凍パスフレーズを更新する．更新が完了したら，証明書ストアサーバに対して更新完了を通知する．
8. PKCS#12 の受け取り
   証明書ストアサーバから送信されてきた，当該 IC カードに対応して登録されている暗号化された PKCS#12 を受け取り，対応する解凍パスフレーズを用いて復号し，当該 PKCS#12 に格納されていた秘密鍵を利用可能にする．（最大で 6 つの PKCS#12 がやりとりされる．）
9. カード PIN の更新
   任意の時点で，利用者からのカード PIN の更新要求に対応する．
10. 秘密鍵の無効化
    IC カードがリーダライタから取り外された場合は，秘密鍵を無効化する．

なお、クライアント証明書（PKCS#12）の更新時は、復

号するために用いる解凍パスフレーズの更新も必要なため，初回アクセスと同様の手順により，IC カード側の解凍パスフレーズの書き換えを行う．また，カード再発行の際は，原則としてクライアント証明書（PKCS#12）も再発行することが望ましいが，引き続き同一の証明書を利用する必要がある場合（S/MIME 等を利用している場合）は，PKCS#12 に対応する解凍パスフレーズを新たな初期解凍パスフレーズで暗号化しなおす必要がある．

6. 今後の展望

UPKI パスの実装は，現時点で Windows に対応しているが，UPKI パスの仕様は汎用リーダライタが利用できることから NFC (Near Field Communication) [22]に対応したスマートフォンやタブレットにも実装可能であると考えられ，UPKI パスの応用範囲はさらに広がる．

実際に UPKI パスを大学等で利用するためには，キャンパス内の ID 管理，FCF キャンパスカードの発行管理およびクライアント証明書の発行管理を連携し，全体にかかる運用管理コストをさらに削減可能な，キャンパス全体の ID 管理フレームワークを設計し実現すべきである．将来的には，必要な機能をパッケージとして備えた製品やクラウドサービスが登場することを期待したい．

また，セキュリティ面についても，さらなる改善が必要である．今回は PKCS#11 や CAPI を利用し，OS 標準の証明書ストアを利用しないことで，ある程度の脆弱性回避を試みたが，同一 OS 上で動作するソフトウェア処理である限り，脆弱性は残る．たとえば，PKCS#11 の API を介してクラウド上の Hardware Security Module (HSM)等の仕組みと連携し，端末側に秘密鍵を持ち込まないようにすることできれば，端末上で秘密鍵が危殆化するような問題は回避することができる．しかし，そのような仕組みを実現するためには，API にアクセスする際の認証を依然として検討する必要がある．

7. おわりに

本稿では，JIPDEC がクライアント証明書の普及のために考案した JCAN パスについて，UPKI クライアント証明書と組み合わせて活用するために，特にセキュリティの観点から仕様の見直しを行った．仕様の見直しを受けて改良したものは UPKI パスと呼ぶ．今回実装したシステムは，これから評価を行う段階であるが，今後，このような，クライアント証明書を効果的に活用する仕組みの展開が進むことで，クライアント証明書の普及が進み，認証のセキュリティが向上することを期待したい．

謝辞　システムの改良にあたり各種検討にご協力頂いた，一般財団法人日本情報経済社会推進協会　大泰司章氏，(株) 高見沢サイバネティックス　増井正宏氏，トッパンフォームズ（株）荻原晴子氏，その他の関係の方々に，謹んで感謝の意を表する．

参考文献

1) Sony Corporation, FeliCa, http://www.sony.co.jp/Products/felica/ (Accessed 13 April 2015).
2) 財団法人日本情報処理開発協会, “電子認証の民間制度・基盤確立に関する調査研究報告書”, 2010. http://www.jipdec.or.jp/project/anshinkan/doc/2009/21-h006_01.pdf
3) 財団法人日本情報処理開発協会, “電子認証の民間制度・基盤の確立に関する調査研究報告書”, 2011. http://www.jipdec.or.jp/pdf/project/jka/2010/22-h003report.pdf
4) ITU-T Recommendation X.509: Information Technology - Open Systems Interconnection - The Directory: Authentication Framework, 2005.
5) Housley, R., W. Ford, W. Polk and D. Solo, “Internet X.509 Public Key Infrastructure: Certificate and CRL Profile”, RFC 2459, 1999.
6) Housley, R., W. Ford, W. Polk and D. Solo, “Internet X.509 Public Key Infrastructure: Certificate and CRL Profile”, RFC 3280, 2002.
7) David Cooper, et al., “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile”, RFC 5280, 2008.
8) B. Ramsdell, S. Turner, “Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification”, RFC5751, 2010.
9) 一般財団法人情報経済社会推進協会, “サイバーID 証明書 JCAN”, http://jcan.jipdec.or.jp/ (Accessed 13 April 2015).
10) 一般財団法人情報経済社会推進協会, “サイバー法人台帳 ROBINS”, http://robins-cbr.jipdec.or.jp/ (Accessed 13 April 2015).
11) 福田昭和, “JCAN ビジネスパス”, 電子認証の民間制度・基盤の確立に関するシンポジウム（財団法人日本情報処理開発協会 電子認証等の民間制度・基盤の確立に関する委員会）, 2010. http://www.jipdec.or.jp/project/anshinkan/doc/20100204/06
12) K. Moriarty, Ed., M. Nystrom, S. Parkinson, A. Rusch, M. Scott, “PKCS #12: Personal Information Exchange Syntax v1.1”, RFC7292.
13) FeliCa 共通利用フォーマット推進フォーラム, “ID カードの共通フォーマットが Ver アップ 安全強化と NFC 対応で時代の要請に応える, CardWave, Jan-Feb 2014, pp. 42-43, 2014. http://www.fcf.jp/PDF/cardwave14_01-02_fcf.pdf
14) 島岡政基, 西村健, 古村隆明, 中村素典, 佐藤周行, 岡部寿男, 曽根原登, “学術機関のためのサーバ証明書発行フレームワーク” (ネットワーク管理・オペレーション, <特集>若手研究者のためのフロンティア論文), 電子情報通信学会論文誌, Vol.J54-B, No.7, pp.871-882, 2012.
15) GÉANT Association, TERENA Certificate Service (TCS), https://www.terena.org/activities/tcs/ (Accessed 13 April 2015)
16) InCommon LLC, InCommon Certificate Service, https://www.incommon.org/certificates/ (Accessed 13 April 2015).
17) Sony, FeliCa 技術方式の各種コードについて”, 2010. http://www.sony.co.jp/Products/felica/business/tech-support/
18) FeliCa 共通利用フォーマット推進フォーラム, “FCF Ver.3 新フォーマットのご紹介”, 2013. http://www.fcf.jp/fcf_ver3/fcf_ver3.pdf
19) G. Zorn, “Microsoft PPP CHAP Extension, Version 2”, RFC2759, 2000.
20) PKCS#11, https://www.oasis-open.org/committees/pkcs11/
21) Microsoft, Cryptography API: Next Generation (CNG), https://msdn.microsoft.com/en-us/library/windows/desktop/aa376210%28v=vs.85%29.aspx
22) NFC Forum, NFC Forum Technical Specifications, http://members.nfc-forum.org/specs/spec_list/ (Accessed 13 April 2015)